



# THE UNIVERSITY *of* EDINBURGH

This thesis has been submitted in fulfilment of the requirements for a postgraduate degree (e.g. PhD, MPhil, DClinPsychol) at the University of Edinburgh. Please note the following terms and conditions of use:

This work is protected by copyright and other intellectual property rights, which are retained by the thesis author, unless otherwise stated.

A copy can be downloaded for personal non-commercial research or study, without prior permission or charge.

This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the author.

The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the author.

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given.

# Wide-Coverage Statistical Parsing with Minimalist Grammars

*John Torr*



Doctor of Philosophy  
Institute for Language, Cognition and Computation  
School of Informatics  
University of Edinburgh  
2019



# Abstract

Syntactic parsing is the process of automatically assigning a structure to a string of words, and is arguably a necessary prerequisite for obtaining a detailed and precise representation of sentence meaning. For many NLP tasks, it is sufficient to use parsers based on simple context free grammars. However, for tasks in which precision on certain relatively rare but semantically crucial constructions (such as unbounded wh-movements for open domain question answering) is important, more expressive grammatical frameworks still have an important role to play.

One grammatical framework which has been conspicuously absent from journals and conferences on Natural Language Processing (NLP), despite continuing to dominate much of theoretical syntax, is Minimalism, the latest incarnation of the Transformational Grammar (TG) approach to linguistic theory developed very extensively by Noam Chomsky and many others since the early 1950s. Until now, all parsers using genuine transformational movement operations have had only narrow coverage by modern standards, owing to the lack of any wide-coverage TG grammars or tree-banks on which to train statistical models. The received wisdom within NLP is that TG is too complex and insufficiently formalised to be applied to realistic parsing tasks. This situation is unfortunate, as it is arguably the most extensively developed syntactic theory across the greatest number of languages, many of which are otherwise under-resourced, and yet the vast majority of its insights never find their way into NLP systems. Conversely, the process of constructing large grammar fragments can have a salutary impact on the theory itself, forcing choices between competing analyses of the same construction, and exposing incompatibilities between analyses of different constructions, along with areas of over- and undergeneration which may otherwise go unnoticed.

This dissertation builds on research into computational Minimalism pioneered by Ed Stabler and others since the late 1990s to present the first ever wide-coverage Minimalist Grammar (MG) parser, along with some promising initial experimental results. A wide-coverage parser must of course be equipped with a wide-coverage grammar, and this dissertation will therefore also present the first ever wide-coverage MG, which has analyses with a high level of cross-linguistic descriptive adequacy for a great many English constructions, many of which are taken or adapted from proposals in the mainstream Minimalist literature. The grammar is very deep, in the sense that it describes many long-range dependencies which even most other expressive wide-coverage grammars ignore. At the same time, it has also been engineered to be highly constrained,

with continuous computational testing being applied to minimize both under- and over-generation.

Natural language is highly ambiguous, both locally and globally, and even with a very strong formal grammar, there may still be a great many possible structures for a given sentence and its substrings. The standard approach to resolving such ambiguity is to equip the parser with a probability model allowing it to disregard certain unlikely search paths, thereby increasing both its efficiency and accuracy. The most successful parsing models are those extracted in a supervised fashion from labelled data in the form of a corpus of syntactic trees, known as a treebank. Constructing such a treebank from scratch for a different formalism is extremely time-consuming and expensive, however, and so the standard approach is to map the trees in an existing treebank into trees of the target formalism. Minimalist trees are considerably more complex than those of other formalisms, however, containing many more null heads and movement operations, making this conversion process far from trivial. This dissertation will describe a method which has so far been used to convert 56% of the Penn Treebank trees into MG trees. Although still under development, the resulting MGBank corpus has already been used to train a statistical A\* MG parser, described here, which has an expected asymptotic time complexity of  $O(n^3)$ ; this is much better than even the most optimistic worst case analysis for the formalism.

# Lay Summary

Human language sentences are sequences of words, but they also have a complex hidden structure. The task of programming machines to automatically determine that structure is known as *syntactic parsing*. This process is a crucial intermediate step towards converting human language into a machine understandable formal language, with the goal of enabling humans to communicate more naturally with computers. Syntactic parsing is a very difficult task because human language turns out to be highly ambiguous in ways of which we humans are very often unaware, being such experts at the parsing task ourselves. To help to overcome this ambiguity issue, modern wide-coverage parsers use statistical models which allow the machine to choose between alternative possible structures based on which is most probable; this approach requires a large corpus of sentences annotated with their structures by humans from which the computer can learn the relevant probability distributions.

A further issue is that there are several competing theories within linguistics about the precise nature of the human grammar and of the structures it generates. Arguably the dominant theory of syntax within theoretical linguistics is Minimalism, which is part of a more general framework known as Transformational Grammar, which Noam Chomsky and others have been developing since the 1950s. This theory is very complex, and partly for this reason has so far largely been avoided by computer scientists.

This dissertation builds upon foundational theoretical research into computational Minimalism pioneered by Ed Stabler and others since the late 1990s, and applies this framework for the first time to the more practical task of wide-coverage statistical parsing, with the aim of showing that these grammars are computable at scale. The dissertation presents the first ever wide-coverage Minimalist grammar and parser along with the tools and corpus data that were created for the project.

*(John Torr)*

# Acknowledgements

A huge thank you must go first and foremost to my PhD supervisor, Mark Steedman, who was a great sport for agreeing to supervise a dissertation on Transformational Grammar parsing. CCG is arguably the most successful of the linguistically expressive formalisms to have been applied to the wide-coverage parsing task, and having the support, guidance and critical eye of its inventor has been invaluable to say the least.

I would also like to thank my second supervisor, Shay Cohen, who was always very helpful and encouraging whenever I sought his guidance.

Ed Stabler also deserves a very big thank you, having served as an unofficial advisor throughout the project. Ed is the inventor of the MG formalism, and so it goes without saying that without him this dissertation would not exist.

A very big thank you must also go to Miloš Stanojević, who constructed the supertagger which the A\* parser described in this dissertation relies on, and whose support and guidance, particularly over the last few months, has been invaluable. Miloš is of a very rare breed indeed: a machine learning expert interested in Chomskyan syntax, and the field of MG parsing is lucky to have him.

I would also like to thank my thesis examiners John Hale and Frank Keller for their very helpful feedback during my viva.

I could also never have undertaken this project were it not for the knowledge of Minimalist syntax which I gained during three incredibly enjoyable years as a linguistics undergraduate. I would therefore also like to thank my undergraduate syntax lecturer and supervisor, Theresa Biberauer, for many hours of fascinating discussions, and for her indefatigable enthusiasm and support throughout that time. I also wish to thank Paula Buttery, who was my undergraduate computational linguistics lecturer and supervisor, and without whose guidance and encouragement (and introduction of the Li.18 computational linguistics course) I never would have ended up pursuing a computational route for my graduate studies.

I am also eternally grateful to Charles Moseley and Mike Franklin for believing in me enough to offer me a place as an undergraduate at Hughes Hall College, an event which has had an immeasurably positive impact on my life.

I would also like to express my gratitude to Ian Baird, at one time the head of the English department at Framlingham College, Suffolk, for freely giving up his spare time to tutor me for the English Language A-level from 1996 to 1998. It was during those lessons that I really discovered my love of grammar (and what the difference

between a subject and an object was), and without Dr Baird's generosity the present thesis would almost certainly never have been written.

I also gratefully acknowledge the Engineering and Physical Sciences Research Council (EPSRC), whose studentship has supported me financially throughout my time as a PhD student.

Finally, I would like to thank my parents for their support (both financial and emotional) throughout my 10 year journey as a university student.

*(John Torr)*



# Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

*(John Torr)*

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Introduction and Motivation . . . . .	2
1.2	The thesis proposed . . . . .	12
1.3	Contributions . . . . .	12
1.4	The structure of this dissertation . . . . .	12
1.5	Table of notational conventions used in this thesis . . . . .	15
<b>I</b>	<b>Competence</b>	<b>19</b>
<b>2</b>	<b>Minimalism, Minimalist Grammars and Transformational parsing</b>	<b>21</b>
2.1	Background . . . . .	22
2.2	Minimalism . . . . .	24
2.3	Minimalist Grammars . . . . .	26
2.4	Parsing with Transformational Grammar . . . . .	31
2.4.1	Early TG parsing . . . . .	31
2.4.2	Augmented Transition Networks . . . . .	32
2.4.3	Parsifal . . . . .	33
2.4.4	Principles-based parsers . . . . .	33
2.4.5	Principar . . . . .	34
2.4.6	Minimalist parsers . . . . .	35
2.4.6.1	Minipar . . . . .	35
2.4.6.2	MG Parsers . . . . .	36
2.4.7	The Minimalist Machine . . . . .	37
<b>3</b>	<b>Extended Directional Minimalist Grammars</b>	<b>39</b>
3.1	Introduction . . . . .	40
3.2	Directional Minimalist Grammars . . . . .	40

3.2.1	A simple derivation using Merge . . . . .	43
3.2.2	Classical Xbar Theory, Derived Xbar Theory and MGs . . . .	45
3.2.3	Phrasal Movement . . . . .	52
3.2.4	A derivation involving Move . . . . .	53
3.2.5	MG derivation trees . . . . .	56
3.2.6	Formal definition of a Directional Minimalist Grammar . . . .	63
3.2.7	Null Heads and Basic Clause Structure . . . . .	65
3.2.8	The tripartite structure of clauses . . . . .	76
3.2.9	Keeping null heads grounded . . . . .	77
3.2.10	Derivational Economy and The Shortest Move Constraint . .	79
3.3	Extended Directional Minimalist Grammars . . . . .	86
3.3.1	Adjunction . . . . .	87
3.3.2	Head Movement . . . . .	92
3.3.3	Rightward Movement . . . . .	95
3.3.4	Covert Movement/Move-F/Agree and !-type suicidal licensors	99
3.3.5	Existentials and Multiple Agree in MGbank . . . . .	110
3.3.6	Raising and Obligatory Control . . . . .	119
3.3.7	Coordination and other related phenomena . . . . .	126
3.3.7.1	The Xbar Theoretic Approach to Basic Coordinate Structures . . . . .	128
3.3.7.2	Across-the-board phrasal movement, adjunct control and parasitic gaps . . . . .	131
3.3.7.3	On sideward movement . . . . .	135
3.3.7.4	Across-the-board head movement . . . . .	136
3.3.7.5	On the coordination of lexical $X^0$ heads . . . . .	139
3.4	Incorporating I-selection and Agreement into MGs . . . . .	142
3.4.1	Case ‘Assignment’ . . . . .	143
3.4.2	Fine-grained Subcategorization . . . . .	149
3.4.3	Agreement . . . . .	152
3.5	Related work on agreement and fine-grained subcategorisation . . . .	153
<b>4</b>	<b>Locality</b>	<b>157</b>
4.1	Introduction . . . . .	158
4.2	Derelativized SMC (DSMC) . . . . .	158
4.2.1	An excursus on restrictive relative clauses . . . . .	160

4.2.2	Empirical motivation and problems for DSMC . . . . .	167
4.2.3	Gerunds and DSMC . . . . .	174
4.2.4	Reflexive Binding, Principle A and DSMC . . . . .	176
4.2.4.1	Across-the-board movement and binding . . . . .	197
4.2.5	Covert-only and overt-only licensees . . . . .	198
4.2.6	Multiple Agree and DSMC . . . . .	199
4.3	Containment-based locality constraints . . . . .	205
4.3.1	Condition on Extraction Domains . . . . .	206
4.3.1.1	The Specifier Island Constraint . . . . .	207
4.3.1.2	The Adjunct Island Constraint . . . . .	211
4.3.2	The Coordinate Structure Constraint . . . . .	214
4.3.3	The Right Roof Constraint . . . . .	216
4.3.4	Phases, Successive Cyclic A'-movement and ?-type suicidal licensors . . . . .	220
4.3.5	The Complex NP Constraint . . . . .	234
4.3.6	Interrogative clause islands . . . . .	235
4.4	Subject/non-subject asymmetries . . . . .	236
4.4.1	Complementizer-trace effects . . . . .	236
4.4.1.1	<i>that</i> -trace effects and property suppressor features . . . . .	236
4.4.1.2	<i>anti-that</i> -trace effects . . . . .	238
4.4.1.3	<i>for</i> -trace effects . . . . .	242
4.4.2	<i>Do</i> -trace effects . . . . .	242
4.5	The time complexity of the formalism . . . . .	243
4.6	The expressive power of the formalism . . . . .	245
4.7	The fine-grained structure of clauses . . . . .	246
4.7.1	The theta domain (VP) . . . . .	246
4.7.2	Case Theory and the Case Filter . . . . .	251
4.7.3	Infinitival clauses and successive cyclic A-movement . . . . .	254
4.7.4	Non-obligatory Control . . . . .	264
4.7.5	The inflectional domain (TP) . . . . .	269
4.7.6	The discourse domain (CP) . . . . .	275

<b>5</b>	<b>Autobank and MGbank: semi-automatically constructing a wide-coverage MG treebank</b>	<b>287</b>
5.1	Introduction . . . . .	288
5.2	Semi-manual construction of treebanks . . . . .	289
5.3	Semi-automatic conversion of an existing treebank into a different formalism . . . . .	289
5.3.1	The transduction approach . . . . .	289
5.3.2	The parser approach . . . . .	292
5.4	Autobank . . . . .	293
5.4.1	Preprocessing . . . . .	294
5.4.2	The manual annotation phase . . . . .	300
5.4.3	The automatic annotation phase . . . . .	304
5.4.4	Automatic scoring of candidate trees . . . . .	306
5.4.4.1	Dependency-based scoring . . . . .	307
5.4.4.2	Other metrics . . . . .	314
5.4.5	Constraining the parser's search space with PTB and CCG-bank constituencies . . . . .	315
5.5	MGbank . . . . .	317
5.5.1	Corpus statistics . . . . .	317
5.6	Evaluation of MGbank . . . . .	318
5.6.1	Global dependency evaluation . . . . .	318
5.6.2	Error analysis of global dependency recovery . . . . .	322
5.6.3	Targeted evaluation of some long-distance dependency constructions in MGbank . . . . .	327
5.6.3.1	Passivization . . . . .	329
5.6.3.2	<i>Want</i> -type subject control . . . . .	330
5.6.3.3	Subject raising verbs . . . . .	332
5.6.3.4	Exceptional Case Marking (raising to object) . . . . .	332
5.6.3.5	Object Control . . . . .	335
5.6.3.6	Control into infinitival purposive clauses (adjunct control) . . . . .	336
5.6.3.7	<i>for-to</i> infinitivals . . . . .	338
5.6.3.8	Infinitival relative clauses (subject and object) . . . . .	339

5.6.3.9	Finite non-reduced subject relative clauses . . . . .	341
5.6.3.10	Finite restrictive reduced passive subject relative clauses	344
5.6.3.11	Free subject relatives . . . . .	345
5.6.3.12	Finite object relative clauses . . . . .	346
5.6.3.13	Free object relatives . . . . .	347
<b>6</b>	<b>Wide-coverage neural A* parsing for Minimalist Grammars</b>	<b>349</b>
6.1	Introduction . . . . .	350
6.2	A* CCG parsing . . . . .	351
6.3	From CKY MG parsing to A* MG parsing . . . . .	353
6.4	Porting L&S's CCG parsing model to MG . . . . .	354
6.4.1	The problem of multiple spans . . . . .	356
6.4.2	The problem of across-the-board movement . . . . .	356
6.4.2.1	An excursus on tough movement . . . . .	357
6.4.3	The problem of null heads . . . . .	362
6.4.4	Adapting an existing CKY MG parser to use MG supertags . .	367
6.5	Experiments . . . . .	368
6.5.1	Model description . . . . .	368
6.5.2	Recovery of MGBank dependencies . . . . .	369
6.5.3	Comparison to CCG . . . . .	372
6.5.4	Parsing speed . . . . .	375
6.5.5	Coverage . . . . .	376
6.5.6	Recovery of unbounded object extraction dependencies . . . .	376
6.5.6.1	Recovery of free object relative dependencies . . .	377
6.5.6.2	Recovery of non-free object relative dependencies .	377
6.5.6.3	Recovery of tough movement dependencies . . . .	378
<b>III</b>	<b>Conclusion</b>	<b>383</b>
<b>7</b>	<b>Conclusion</b>	<b>385</b>
7.1	Summary of contributions . . . . .	386
7.2	Future Work . . . . .	389

<b>IV</b>	<b>Appendices</b>	<b>393</b>
<b>A</b>	<b>A selection of constructions and their lexical entries in MGbank</b>	<b>395</b>
A.1	Introduction . . . . .	395
A.2	Basic n-place predicate constructions . . . . .	396
A.2.1	little v heads . . . . .	396
A.2.2	Unergative intransitive verbs . . . . .	396
A.2.3	Unaccusative verbs . . . . .	396
A.2.4	Transitives . . . . .	397
A.2.5	Ditransitives . . . . .	397
A.2.6	Prepositional datives . . . . .	397
A.2.7	Reporting verbs . . . . .	397
A.2.8	4-place verbs . . . . .	398
A.2.9	The copula . . . . .	398
A.2.10	'Restructuring' verbs taking direct vP or VP complement . . .	399
A.3	Passives and expletive <i>it</i> . . . . .	399
A.3.1	Passive auxiliaries . . . . .	399
A.3.2	Passivised transitive verb . . . . .	399
A.3.3	Expletive Passive/Progressive . . . . .	399
A.3.4	Impersonal passives . . . . .	400
A.3.5	<i>it</i> -extraposition . . . . .	400
A.3.6	Pseudo Passive . . . . .	401
A.4	Particle verbs . . . . .	401
A.5	The inflection domain . . . . .	402
A.5.1	Progressive auxiliaries . . . . .	402
A.5.2	Perfective auxiliaries . . . . .	402
A.5.3	Negation . . . . .	402
A.5.4	Modal verbs . . . . .	403
A.5.5	Subjunctive Mood . . . . .	403
A.5.6	Tense . . . . .	404
A.5.7	Emphatic <i>do</i> . . . . .	404
A.6	More A-movement constructions . . . . .	405
A.6.1	subject raising verbs . . . . .	405
A.6.2	Raising adjectives . . . . .	405
A.6.3	Exceptional Case Marking (ECM) (object raising) . . . . .	405

A.6.4	<i>want</i> -type subject control . . . . .	405
A.6.5	<i>promise</i> -type subject control . . . . .	405
A.6.6	Object control . . . . .	406
A.6.7	Adjunct control . . . . .	406
A.6.8	Non-obligatory control . . . . .	406
A.6.9	<i>for-to</i> infinitivals . . . . .	406
A.6.10	Floating quantifiers . . . . .	407
A.6.11	Locative Inversion . . . . .	407
A.6.12	Expletive <i>there</i> . . . . .	408
A.6.13	Clausal subjects . . . . .	408
A.6.14	Reflexive and Reciprocal anaphors . . . . .	408
A.7	Sentential force . . . . .	409
A.7.1	Declaratives . . . . .	409
A.7.2	Imperatives . . . . .	409
A.7.3	Exclamatives . . . . .	410
A.7.4	Conditionals . . . . .	410
A.7.5	Interrogatives . . . . .	410
A.7.6	Echo questions . . . . .	411
A.8	Adverbs . . . . .	412
A.9	More A'-movement . . . . .	412
A.9.1	Relative clauses . . . . .	412
A.9.2	Clefts . . . . .	414
A.9.3	Pseudo-clefts . . . . .	415
A.9.4	Topicalization . . . . .	415
A.9.5	Quotative inversion and discontinuous quotatives . . . . .	415
A.9.6	Negative Inversion . . . . .	416
A.9.7	Heavy DP shift . . . . .	416
A.9.8	Tough movement . . . . .	416
A.9.9	The violin-sonata paradox . . . . .	417
A.9.10	Pied-piping heads . . . . .	418
A.10	The nominal domain . . . . .	419
A.10.1	Nouns, names and honorifics . . . . .	419
A.10.2	Determiners . . . . .	420
A.10.3	Pronouns . . . . .	421
A.10.4	Possessives . . . . .	422



A.10.5	Prepositions . . . . .	423
A.10.6	Gerunds . . . . .	423
A.11	Adjectives . . . . .	424
A.12	Coordination . . . . .	424
A.12.1	Basic coordinators . . . . .	424
A.12.2	ATB head movement and argument cluster coordinators . . . . .	425
A.12.3	Correlative focus coordinator particles . . . . .	425
<b>B</b>	<b>The grammar rules</b>	<b>427</b>
B.1	Basic Merge and Move rules from section 3.3.2 . . . . .	427
B.2	Adjunction rules from section 3.3.1 . . . . .	428
B.3	Head movement rules from section 3.3.2 . . . . .	429
B.4	Rightward Move rule from section 3.3.3 . . . . .	429
B.5	Covert Move rules from section 3.3.4 . . . . .	430
B.6	Suicidal licenser rules from section 4.3.4 . . . . .	431
B.7	Control Move rules from section 3.3.6 . . . . .	431
B.8	Across-the-board phrasal movement rules from section 3.3.7.2 . . . . .	432
B.9	Across-the-board head movement rules from section 3.3.7.4 . . . . .	432
B.10	The rule allowing for list coordination from section 3.3.7.1 . . . . .	433
B.11	Lexical head coordination rules from section 3.3.7.5 . . . . .	433
B.12	Specifier escape rules from section 4.3.1.1 . . . . .	434
<b>C</b>	<b>The syntactic and semantic PTB head finding rules</b>	<b>435</b>
C.1	Head-finding rules for PTB trees . . . . .	435
C.1.1	Syntactic head-finding rules . . . . .	435
C.1.2	Semantic head-finding rules . . . . .	437
<b>D</b>	<b>The 100 sentences used for the global dependency evaluation of MGbank</b>	<b>439</b>
D.1	The gold test set for global evaluation of MGbank . . . . .	439
	<b>Bibliography</b>	<b>443</b>

# **Chapter 1**

## **Introduction**

## 1.1 Introduction and Motivation

*A goal of earlier linguistic work, and one that is still a central goal of the linguistic work that goes on in computational linguistics, is to develop grammars that assign a reasonable syntactic structure to every sentence of English...This is not a goal that is currently much in fashion in theoretical linguistics [where] the development of large fragments has long since been abandoned in favor of the pursuit of deep principles of grammar.*

(Abney, 1996, page 20), cited in Müller (2016)

The topic of this dissertation is wide-coverage parsing with Minimalist Grammars (MGs). MGs were introduced in Stabler (1997) and are a computationally oriented and rigorous formalisation of many aspects of Chomsky's Minimalist Program (MP), particularly in its early (Chomsky, 1995) iteration before the introduction of phases and the probe-goal/Agree framework. Minimalism itself is the latest incarnation of the highly influential Transformational Grammar (TG) approach to linguistic theory developed very extensively by Noam Chomsky and others since the 1950s.

While MG parsers have existed since Harkema (2001), so far these have all been small-scale theoretical implementations - essentially proofs of concept equipped only with toy grammars. This of course entails that until now there have also been no wide-coverage MG treebanks on which to train efficient statistical MG parsers. There appear to be two primary reasons for this. The first is that Minimalism, and more broadly TG, is a very complex syntactic theory with a vast literature containing many competing proposals for most construction types; this makes the task of navigating this literature - while filling in the blanks for certain linguistically uninspiring but empirically ubiquitous constructions not covered in the literature - in order to formalise a concrete wide-coverage grammar for a particular language, very challenging.

The second reason is that the computational work so far conducted in the MG framework has been largely oriented towards computational psycholinguistics (Hale, 2001, 2003, 2011; Kibele et al., 2013; Graf and Marcinek, 2014; Gerth, 2015; Graf et al., 2015b; Brennan et al., 2016; Graf et al., 2017; Hale et al., 2018), rather than towards NLP, where the construction of wide-coverage grammars and treebanks is today a *sine qua non*. In particular, owing to their close affinity with what arguably remains the dominant framework in theoretical syntax, MGs have proven a popular choice for investigations aiming to explain certain processing effects for particular constructions relative to: 1. different competing theoretical proposals about the precise nature of the syntactic structures involved (i.e. about linguistic *competence*, or the speaker's knowledge of language); and 2. different proposals about the nature of the human parser (i.e.

about the speaker's processing of language, which is one aspect of linguistic *performance*<sup>1</sup>). A classic example of such an effect which has featured prominently in this literature is the fact that humans find certain types of relative clause (e.g. subject vs object relativization) more difficult to process than others. In such a context, it is feasible to model just the specific construction(s) under investigation, and to abstract away from certain details of performance, such as the question of how the parser selects the correct parse from among the often huge range of possible alternatives.

To give an example, Graf et al. (2017) investigate a wide array of potential metrics relating memory usage to the difficulty humans experience when parsing relative clauses, under the assumption that the human parser operates along the lines of the top-down incremental algorithm described in Stabler (2013b). These metrics are intended to provide an empirical means by which to evaluate competing Minimalist proposals for the structure of relative clauses. An integral part of Stabler's parser is a probabilistic beam for pruning the search space, the proper implementation of which would require a wide-coverage corpus of MG derivation trees from which to extract realistic parameters. However, in the following quotation, Graf et al. (2017) are explicit about their intention to abstract away from the search problem.

*The most demanding task of parsing - searching through a large space of structures in the search for the correct one - is taken out of the equation...this does not deny that ambiguity has a large role to play, e.g. in garden path sentences, but it is taken out of the equation in order to determine the relevance of isolated structural factors. This simplification is shared among all recent work that use Stabler's MG parser to model human processing.*

(Graf et al., 2017, pages 70-71)

Just as Chomsky (1965) argued for the utility of abstracting away from performance when studying competence, these authors are here making a similar case for abstracting away from one aspect of performance (i.e. the non-deterministic properties of the parser) to isolate another aspect of performance (i.e. the algorithmic properties of the parser; here, its top-down, incremental nature), with the aim of 'see[ing] how much of human sentence processing can be explained by considering only the order of how the parts of the correct derivation are built' (Graf et al., 2017, page 70).

While such abstraction can certainly be a very effective methodological tool, it is of course, as Graf et al. (2015b, page 3) acknowledge, ultimately an implausible idealization. Just as a complete linguistic theory will ultimately need to incorporate a theory

---

<sup>1</sup>See Chomsky (1965) on the competence-performance distinction in linguistics.

of both competence and performance,<sup>2</sup> so too will a complete theory of performance need to incorporate both the non-deterministic and algorithmic aspects of parsing.

There is in fact good evidence that, alongside structure and memory requirements, ambiguity is itself an important predictor of the difficulty with which humans process different types of relative clause. Hale (2003) shows that this degree of difficulty (measured for instance by on-line reading times) correlates well with the amount of entropy reduction which must be performed during parsing, and that this correlation is stronger if Kayne's (1994) promotion analysis of relative clauses is adopted rather than a more traditional adjunction-based account. Hale's study used a toy probabilistic MG and an artificial corpus of 24 sentences containing 6 types of relative clause. Each sentence was weighted by the frequency of the relative clause it contained according to a manual corpus study by Keenan and Hawkins (1987). This was done in order 'to make their frequencies more realistic' (Hale, 2003, section 3.6). It seems likely, therefore, that this type of psycholinguistic study could benefit from the availability of an (easily modifiable) wide-coverage corpus of MG derivation trees from which to extract much *more* realistic probability distributions.

While MG research has so far focused primarily on characterising the grammar and the parser, there have also been some preliminary investigations into grammar induction with this formalism (Stabler, 1998; Bonato and Retoré, 2001; Portelance et al., 2017). A central idea in modern TG is that all language variation is encoded in the lexicon (Borer, 1984; Chomsky, 1995; Baker, 2008) (perhaps isolable to the functional heads) with the rules of the grammar being fixed and universal. Acquiring a grammar for a particular language then amounts to acquiring its lexicon, certain aspects of which (such as the order of different heads along the clausal spine (Cinque, 1999; Rizzi, 1997)) may also be fixed in advance.

Even with parts of the grammar innately prespecified, under reasonable assumptions it will still be the case that many grammars are compatible with the input that a language learner has encountered at any given point. Some evaluation metric must therefore be determined in order to allow the learner to choose between competing

---

<sup>2</sup>Chomsky appears to acknowledge this in the following quotation (cited in Hale (2003)):

The I-language is a (narrowly described) property of the brain, a relatively stable element of transitory states of the language faculty. Each linguistic expression (SD) generated by the I-language includes instructions for the performance systems in which the I-language is embedded. *It is only by virtue of its integration into such performance systems that this brain state qualifies as language.* [emphasis added]

(Chomsky, 1992, page 213)

grammar hypotheses. Pursuing ideas which go back to Chomsky (1965) and were developed further in Berwick (1982, 2015), Li and Vitányi (1992), Rissanen and Ristad (1994) and elsewhere, Stabler (1998) suggests a *Minimal Description Length* approach to acquiring MGs, under which the learner narrows their search space by inherently preferring more succinct grammars that can generalise better over the data over less succinct ones which generalise less well. However, as Stabler demonstrates, increased generalisation often comes at the cost of a greater number of possible structures assignable to any given string by the grammar (and hence, in information theoretic terms, an increase in the number of bits required to recognise a given string), and so learners must balance these two considerations, attempting to minimise both grammar complexity *and* data description complexity. Stabler (1998) discusses these ideas in relation to a small toy grammar and data set, but he also emphasises the importance of the learner's 'sensitivity to quantity of evidence' and robustness to 'noise' (Stabler, 1998, page 91). It is therefore likely that a wide-coverage MG (preferably one which is easily modifiable so as to accommodate the different assumptions made by each researcher about the grammar), together with a corpus of MG derivation trees, could benefit this kind of acquisition-based research: the grammar would provide a much more realistic final state target for any proposed induction algorithm, while the data would enable researchers to better model the sorts of latent structural regularities and noise to which language acquirers are apparently sensitive.

A wide coverage MG also has potential benefits for more formally oriented MG research. For instance, Graf et al. (2015a) propose a single movement normal form for MGs which 'simplifies the formalism and reduces the complexity of movement dependencies' (Graf, 2018, page 23), but could potentially result in a blow up in the size of the grammar. Graf (2018) argues that, at least with respect to 'movement with a clear function,' (Graf, 2018, page 31) no such blow up obtains for MGs with commonly assumed constraints on movement. However, Graf also observes that,

*this does not guarantee that a wide-coverage MG can be safely translated into SMNF without a significant increase in grammar size...MGs, just like the Minimalist literature they are modeled after, also posit more abstract features...whose only purpose is to produce the observed surface order. These features were completely ignored in this paper because it seems unlikely that manual analysis can reveal much about them. Instead, it seems more promising to run simulations where realistic MGs are automatically converted to SMNF.*

(Graf, 2018, pages 31-32)

This dissertation applies MGs to the task of realistic, wide-coverage parsing for the

first time, presenting the first ever wide-coverage Minimalist Grammar, MG treebank and statistical MG parser. As well as enabling probabilistically-based evaluation of MGs as plausible models of human linguistic competence, these new resources also open up the possibility of using these grammars within applied Natural Language Processing (NLP).


Currently, the most widely used approaches to syntactic parsing within NLP rely on neural probabilistic models which are not defined over explicit formal grammar rules at all (e.g. there is no probability assigned to the rule  $S \rightarrow NP VP$ ), though they are still trained in a supervised fashion on treebanks; instead the rules are encoded only implicitly by the neural model (e.g. as a probability distribution over individual parsing actions, or transitions). This is true whether the output representations are constituencies (Vinyals et al., 2015; Dyer et al., 2016) or word-word dependencies (Dyer et al., 2015; Dozat and Manning, 2017). These grammars which these models implicitly encode are also usually only context free, with the traces of movement in the PTB simply being ignored.

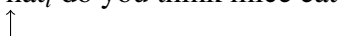
These approaches are very interesting, and have certainly been very successful, at least as far as overall dependency evaluations are concerned. However, when relatively high precision on particular types of (often infrequently occurring but semantically crucial) linguistic construction is important, deep and explicit formal grammatical approaches arguably still have an important role to play. This is particularly true of constructions which are quite rare in corpora and hence difficult for purely statistical approaches to accurately model. It is even more true for constructions which resist a simple context free treatment. Such constructions include unbounded long-distance dependencies, such as *wh* extractions, whose accurate recovery is vital for open-domain question answering systems, for instance: mistaking queries such as *what do you think eats mice? (cats)* with *what do you think mice eat? (cheese)* can quickly lead to user frustration, even if this type of error only occurs once every few hundred queries (at the time of writing, the Google Assistant app interprets both questions as querying the culinary habits of mice; Siri does not even attempt an answer).

Rimell et al. (2009) found that on an unbounded dependency test set, two linguistically expressive parsers, the C&C CCG parser and the Enju HPSG parser, significantly outperformed a number of less expressive parsers (including the popular Stanford Constituency parser) which merely approximate human grammar with finite state or context free covers. Nivre et al. (2010) subsequently found that their purely statistical transition-based and graph-based dependency parsers, when augmented with post pro-

processors for recovering unbounded dependencies, performed only slightly worse than the C&C and Enju parsers on average across all unbounded dependency types. However, on certain constructions, such as those involving object extractions, the difference was more pronounced: the C&C parser's accuracy on non-reduced, non-free object relative clause dependencies was 59.3%, for instance, compared with 40.7% for the highest scoring dependency parser.

A defining property of both Minimalism and the MG formalism modelled after this framework is the use of transformational movement operations to capture many long-distance dependencies. For our earlier examples, for instance, *what* would begin the derivation in the deep subject or object position of the lower clause, before moving to its surface position in the left periphery of the matrix clause, as illustrated in 1 and 2.

(1) What<sub>*i*</sub> do you think t<sub>*i*</sub> eats mice?  


(2) What<sub>*i*</sub> do you think mice eat t<sub>*i*</sub>?  


Although Minimalism continues to dominate much of theoretical linguistics, it currently enjoys far less popularity within computational linguistics, and has until very recently been conspicuously absent altogether from conferences and journals on natural language processing (NLP).<sup>3</sup> One reason for this is that, as Abney observes in the opening quotation to this chapter, the focus within theoretical linguistics (by which he was referring primarily to TG) has been on depth of coverage and the search for universal principles of grammar, rather than on creating holistic wide-coverage grammars for entire languages.

This approach has undoubtedly yielded many important insights, but it has also tended to alienate the mainstream computational linguistics community, for whom the development of wide-coverage grammars is a key concern. The received wisdom within NLP is that TG is too complex and insufficiently formalised to be applied to realistic parsing tasks. This situation is unfortunate, as TG is arguably the most extensively developed syntactic theory across the greatest number of languages, many of which are otherwise under-resourced, and yet the vast majority of its insights never find their way into NLP systems. Conversely, the process of constructing large grammar fragments and subjecting those fragments to computational testing can have a salutary impact on the theory itself, forcing choices between competing analyses of the same construction, and exposing incompatibilities between analyses of different

<sup>3</sup>For an anti-Chomskyan perspective on why this disconnect came about, see Pullum (2009).



constructions, along with areas of over- and undergeneration which may otherwise go unnoticed (Bierwisch, 1963; Abney, 1996) (both cited in Müller (2016)).

A primary goal of the current project was to show that it is possible to construct a formal, wide-coverage grammar of English that is very much in the spirit of mainstream Minimalist proposals. Doing so is far from trivial, however, given that the body of TG literature is vast, meaning that there are usually several alternative analyses to choose from for any given construction type, with different analyses of different constructions by different authors not always fully compatible with one another. Fortunately, the computationally-oriented MG formalism, pioneered by Stabler and others since the late 1990s, provided a solid and rigorously formal foundation for the project. While the MG formalism (and the grammar presented here) differs from MP on certain details, it is sufficiently close to be regarded as a formalisation of the essentials of Chomskyan Minimalism (Müller, 2016, pages 165-166). However, as was already noted, what has so far differentiated MGs from other linguistically expressive formalisms such as TAG, CCG, LFG and HPSG, is that until now there have been no wide-coverage grammar fragments constructed for this framework.

The current project addresses this deficit, presenting the first ever wide-coverage MG in chapters 3 and 4, and in Appendix A. The grammar was manually constructed and subjected to continuous computational testing to check for over- and undergeneration using a piece of grammar development software called Autobank that was itself developed specifically for this project. The development process was corpus-oriented in the sense that it involved the re-annotation of trees from the Penn Treebank (PTB; Marcus et al. 1993) as Minimalist trees. The grammar has been engineered to have a level of (cross-linguistic) descriptive adequacy on a wide range of constructions, including both those which are of particular interest to linguists and those which are found in realistic corpora like the PTB. At the same time, it was engineered to be highly constrained, so that it also blocks many ‘unacceptable’ derivations.<sup>4</sup>

It is important to note that the goal of this project was not to create a grammar with wide enough coverage to parse colloquial dialects, ‘Twitterese’, the English spoken by non-native learners of English or any other non-standard types of English. Instead, the aim was to construct a grammar which can provide precise and linguistically sophisti-

---

<sup>4</sup>As Chomsky (2008, page 11) points out, there are many varieties (or degrees) of unacceptability or grammatical ‘deviance’. However, throughout this dissertation we will make the simplifying assumption that the constraints of the formal grammar are hard, and that they therefore strictly sort sentences (or rather, derivations) into those which are generated and those which are not. Additional soft/gradient constraints can also be enforced by the probabilistic model, of course.

cated analyses of ‘grammatical’ Standard English sentences, including those found in the Wall Street Journal, but also many rare but linguistically more interesting construction types. While finding ways to process other dialects of English is an important task in NLP, restricting our attention to Standard English makes this extremely challenging task far more manageable than would otherwise be the case. It must nevertheless be conceded that this is an unrealistic idealisation from a cognitive perspective, given that humans are able to process all of these different dialects with such apparent ease.

It may seem to be a somewhat futile exercise, at least from the perspective of applied NLP, to spend time modelling constructions which are in some cases so rare that they do not show up in the one million words of text of the Penn Treebank on which most statistical parsing systems are trained.<sup>5</sup> However, ultimately machines will need to be able to properly process the full range of construction types found in natural language if they are ever to attain human-like levels of linguistic competence, and given that pure machine learning approaches tend to fail precisely in those cases where data is most sparse, the development of deep formal models which can correctly analyse and generalise from the rare constructions in the Zipfian tail remains an important long-term goal for NLP.

In order to ensure both reasonably wide and descriptively adequate coverage, it was necessary to synthesise many of the extensions that have been proposed in the MG literature since Stabler’s original formulation. These include the approach to adjunction of Frey and Gärtner (2002), Stabler’s (2001b) formalisation of head movement, and Kobele’s (2008) approach to across-the-board phrasal movement. In addition, a number of novel extensions to the formalism were developed and are presented here, such as mechanisms for across-the-board head movement and lexical head coordination. Finally, in order to keep the grammar very tightly constrained, a novel mechanism is introduced to capture morphosyntactic agreements and fine-grained selectional restrictions which bears some resemblance to the unification approach adopted by formalisms such as HPSG and LFG, but which is flatter and simpler and hence much easier to read and to annotate at speed than the unbounded feature-value matrices used by those formalisms.

Although a strong, precise formal grammatical model can help to restrict the number of analyses entertained by the parser, it is not by itself sufficient for this pur-

---

<sup>5</sup>One example of a construction which does not show up in the PTB but is modelled by the MG grammar presented here is *promise*-type subject control, as in *Jack promised Mary to help*; a further example is the so-called *parasitic gap* construction, as in *which books did you return to the library without reading?*

pose. Natural language sentences are highly ambiguous, locally (i.e. at the morpheme/word/phrase level) as well as globally (at the sentence level), with humans clearly relying on vast amounts of world knowledge alongside their implicit grammatical knowledge in order to efficiently and accurately navigate the search space during parsing. Common sources of ambiguity include lexical ambiguity (*bank* could refer to a financial institution, to the act of keeping one's money in a certain financial institution, to a slope beside a river, and so on), and coordination and prepositional phrase attachment ambiguities. For example, consider the sentence in 3.

(3) Jack rode the bike with no hands

Grammatically speaking, there is no reason why the prepositional phrase *with no hands* could not attach to the nominal phrase headed by *bike*, yielding an interpretation in which Jack rode a bike and that bike had no hands (cf. *Jack rode the bike with no brakes*). A human interlocutor, however, would (under normal contextual circumstances) disregard this structure in favour of the one in which the prepositional phrase modifies the verb phrase headed by *rode*. To do this, we humans rely on our knowledge of the world - in this case, on the knowledge that it is people and not bikes which typically have hands.

Computers currently do not experience the world in the same way or to the same extent as humans, and existing knowledge bases are woefully insufficient for performing the disambiguation task. However, the seminal work of Collins (1999) and Charniak (2000) demonstrated that lexicalised statistical models extracted from a corpus of gold-standard parse trees could be used as a proxy for such world knowledge, enabling machines to perform the disambiguation task reasonably effectively, at least for those constructions which occur relatively frequently in corpora. As well as enabling machines to resolve global ambiguities, such as that exemplified in 3, and thereby hone in on just the correct analysis, statistical models can also enable a parser to reduce local ambiguities which give rise to search paths that may ultimately lead nowhere but which will nevertheless cause the parser to waste valuable computational resources exploring them.

Today, a statistical model component is a virtual necessity for any wide-coverage parsing system, and the best performing statistical parsers are supervised systems, meaning that they are trained on labelled data. For this reason, alongside the grammar, this dissertation will present the first wide-coverage corpus of MG parse trees, known as MGbank. MGbank was generated semi-automatically using the same Auto-

bank software that was used to construct the grammar. The grammar development and treebanking procedures are described in chapter 5 of this dissertation.

Although they already have reasonable coverage, the grammar and treebank remain a work in progress and will be for some years to come. Nevertheless, the corpus has already been used to train the first ever wide-coverage statistical MG parser. This is presented and evaluated against a near-state-of-the-art A\* CCG parser in chapter 6. The MG parser is a variant of the A\* parser for CCG presented in Lewis and Steedman (2014a) and Lewis et al. (2016). Like all modern CCG parsers, this algorithm relies on a statistical supertagger to assign lexical categories to the words of a sentence. However, existing supertaggers can only tag what they can see, which poses a problem for MGs which (following mainstream Minimalist theory) allow for phonetically null syntactic heads. Chapter 6 therefore also presents a novel technique for anchoring null heads to overt ones inside complex LTAG-like MG supertag categories, which allows the statistical supertagging model to be factored over the null heads as well as the overt ones. The parser has a practical expected asymptotic time complexity of  $O(n^3)$ , which is much better than the most optimistic worst case complexity result in the literature, which for MGs with head movement is  $O(n^{2k+5})$  (Stanojević, 2019), where  $k$  is the size of the set of licensees, and equals 4 in the grammar presented here once a derelativized version of SMC that was implemented is taken into account (see section 4.5).

The term *wide-coverage* is of course relative; this parser is certainly by far the widest-coverage MG parser ever constructed, although it currently lags behind wide-coverage parsers for other formalisms at present, returning parses in an ‘Abstract’ mode for 80.6% of sentences in section 00 of the PTB. Its performance on the recovery of both general and unbounded long-distance dependencies is also presently below that of the CCG parser it was compared against. Nevertheless, the results of this first attempt at wide-coverage Minimalist parsing are promising in view of the greater complexity of the underlying Minimalist syntax. CCG already has a very long history in wide-coverage parsing, and the A\* algorithm used here was originally designed with that formalism in mind, so it is perhaps unsurprising that the MG parser’s performance is not yet seriously competitive. The difference should narrow as better models are developed for the MG formalism and as the size and quality of the MGbank corpus increases.

## 1.2 The thesis proposed

The primary thesis of this dissertation is that the MG formalism can (with certain extensions) be used to build deep and precise yet (relatively) efficient wide-coverage statistical parsing systems which are able to directly incorporate many of the proposals from the mainstream Minimalist literature. A further thesis is that constructing wide-coverage Minimalist grammars and subjecting them to computational testing can have a salutary effect on Minimalist theory itself, as this process necessitates rigorous formalisation, reveals incompatibilities between analyses of different constructions, and exposes areas of under- and overgeneration which may otherwise go unnoticed.

## 1.3 Contributions

The primary contributions of the research documented in this dissertation are:

- A number of extensions to the MG formalism, including mechanisms for across-the-board head movement, lexical head coordination and subcategorization/agreement.
- A semi-automatic annotation tool for developing MG grammars and treebanks.
- The first wide-coverage grammar for the MG formalism.
- The first wide-coverage MG treebank on which to train statistical models.
- A novel technique for factoring null heads out from MG parsing which renders MGs fully compatible for the first time with highly efficient Markovian supertagging techniques.
- The first wide-coverage MG parser, which is joint work with Miloš Stanojević. The parser incorporates a statistical model component which was trained, tested and evaluated using MGbank.

## 1.4 The structure of this dissertation

The core of the dissertation is divided into parts 1 and 2. Part 1 is concerned primarily with the competence grammar, although chapter 2 does include a review of some previous TG parsers. Part 2 concerns performance, describing the method used to construct the wide-coverage grammar and treebank, and then presenting the parser itself.

This part is reasonably self contained, with references to the relevant sections in part 1 where required, so that the reader could skip straight to part 2 (and even straight to chapter 6) if parsing is their primary interest. Part 3 concludes the thesis.

Below is a brief summary of the contents of each chapter.

### **Part I: Competence**

**Chapter 2** gives a brief introduction to Minimalism and Minimalist Grammars, before surveying previous work on parsing with Chomskyan syntax.

**Chapter 3** presents the Extended Directional Minimalist Grammars (EDMG) formalism that was constructed for this thesis, and includes detailed discussions of many of the theoretical choices which were made for various constructions (lexical entries for a wider range of constructions in MGBank, along with brief discussions and examples, can be found in Appendix A; a full list of all the grammar rules introduced in this chapter is given for ease of reference in Appendix B).

**Chapter 4** continues the discussion of the grammar, showing how many of the locality constraints from the mainstream Minimalist literature were implemented in the MGBank grammar. This chapter also provides analyses of the formalism's time complexity and expressive power.

### **Part II: Performance**

**Chapter 5** provides an overview of the Autobank system that was developed for this project and then used to construct both the wide-coverage grammar and the MGBank corpus. Some corpus statistics and an evaluation of MGBank in its current iteration are also provided.

**Chapter 6** presents the first ever wide-coverage MG parser, along with an evaluation of its current speed, coverage and accuracy on recovering both general and unbounded dependencies.

### **Part III: Conclusion**

**Chapter 7** concludes the dissertation and suggests some possible directions for future research.

## 1.5 Table of notational conventions used in this thesis

+ (used outside curly braces)	indicates a licensor feature (e.g. +CASE, +wh)
+ (used inside curly braces)	indicates a selectional requirement, e.g. +3SG, +FIN, +FEM, +[NOM ACC] etc
- (used outside curly braces)	indicates a leftward movement licensee feature (e.g. -case, -wh)
- (used inside curly braces)	indicates a negative selectional requirement, e.g. -3SG, -FIN, -FEM, -[NOM ACC] etc
+CASE, +WH etc	uppercased licensors trigger overt movement
+case, +wh etc	lowercased licensors trigger covert movement
d	lowercased d indicates a determiner selectee which cannot persist
D	uppercased D indicates a determiner selectee which can persist after being checked (for control)
+CASE, +WH etc	uppercased licensors trigger overt movement
+case, +wh etc	lowercased licensors trigger covert movement
?	indicates a suicidal licensor which does not delete the licensee it checks (e.g. +wh? used for successive cyclic wh-movement)
!	indicates a suicidal licensor which deletes the licensee it checks (e.g. +self!)
~	(used outside of curly braces) indicates a rightward movement licensee feature, e.g. t~
~	(used inside curly braces) indicates a selectional suppressor, e.g. ~NOM



=	indicates a selector feature (=x indicates leftward selection; x= indicates rightward selection)
≈	indicates an adjunction selector feature (≈x indicates leftward selection; x≈ indicates rightward selection)
^	appears on selector features (e.g. v=^) and triggers head movement with excorporation (for complement selection) or ATB head-movement (for specifier selection)
>	appears on selector features (e.g. >v=) and triggers head movement to the immediate left of the selecting item's head string
<	appears on selector features (e.g. v<=) and triggers head movement to the immediate right of the selecting item's head string
·	separates deleted features from undeleted ones in the lexical head coordination rules
*	equivalent to · above but marks a type saturated constituent
::	type separator indicating a (non-coordinator) underived item (i.e. a lexical item)
:	type separator indicating a (non-coordinator) derived item
;	type separator indicating a item unspecified as derived or underived
:::	type separator indicating an underived coordinator
:::	type separator indicating a derived coordinator

{ and }	surrounds the selectional properties, requirements and restrictions associated with a structure building feature, e.g. +CASE{+ACC.+3SG}
.	a conjunctive separator for selectional properties, requirements and variables, e.g. 3SG.ACC.-FEM.x means ‘has properties 3SG and ACC, and requirement -FEM and variable x’
	indicates inclusive disjunction for selectional properties and requirements. E.g. [+NOM.+ACC] (also written as +[NOM.ACC]) means ‘has NOM or ACC or has both ACC and NOM’, while [+OVERT -OP] means ‘has OVERT or does not have -OP, or has OVERT <i>and</i> does not have OP’
x, y, z	used inside curly braces, indicate selectional variables for percolating selectional properties and requirements
,	(used to the left of the type separator) separates the head string from its left and right dependent strings
,	(used to the right of the type separator) separates the component chains of an expression
$\lambda$	indicates a trace of phrasal movement
$\Lambda$	indicates a trace of head movement
$\mu$	indicates a landing site for covert movement
$\zeta$	indicates a trace of rightward movement

/xxx/	(used in MGbank) lowercased terminals surrounded by slashes indicate terminals which have been rightward moved (hence have phonetic but no semantic content)
/XXX/	(used in MGbank) uppercased terminals surrounded by slashes indicate terminals which have undergone covert movement (and therefore have both semantic and phonetic content)
$\epsilon$	the empty string
[xxx]	used as the string component for underived null lexical items, e.g. [trans], [past] etc (equivalent to $\epsilon$ but easier to read on derivation trees)

Table 1.1

# **Part I**

## **Competence**



## **Chapter 2**

# **Minimalism, Minimalist Grammars and Transformational parsing**

## 2.1 Background

The Minimalist Program (MP) is the latest incarnation of the Transformational Grammar (TG) approach to linguistic theory developed very extensively by Noam Chomsky and others since the 1950s. From its inception, TG has been primarily concerned with explaining how children are able to acquire language with such apparent ease and uniformity in the face of what Chomskians argue to be greatly impoverished linguistic input data. Chomsky (1986b) dubs this *Plato's Problem*. The Chomskyan position has always been that children must approach this task with an innate language faculty, also known as Universal Grammar (UG), which delineates the space of possible human languages and thus simplifies the acquisition task. This implies that all languages are, at some level of abstraction, variations on a single theme, perhaps sharing a common abstract level of structural representation. The task of the linguist is then to provide an accurate characterisation of the content of UG. This is done by first identifying various (often quite subtle) descriptive generalisations across different constructions and languages, and then formulating deep principles of grammar to explain them as elegantly and naturally as possible.

In the Standard Theory (Chomsky, 1957) and its subsequent extensions (Chomsky, 1965) and revisions (Jackendoff, 1977), a context free base component is first used to generate a deep structure for a sentence, after which a cascade of construction-specific transformational rules are applied to map this deep structure into a final surface structure. For example, the rule PASSIVE mapped from an underlyingly active structure to a passive one, causing (among other things) the deep object to be promoted to the subject position. The most important development during the Revised Extended Standard Theory of the 1970s was perhaps the emergence of the abstract Xbar Theoretic approach to phrase structure according to which all phrases share certain commonalities such as headedness and the ability (at least in principle) to take various types of (complement, specifier and adjunct) dependents.

The Revised Extended Standard Theory was replaced in the 1980s by Government and Binding Theory (Chomsky, 1981, 1986a). GB retained a revised version of Xbar Theory, but sought to capture a much greater number of generalisations across less obviously related construction types, and also to capture many more cross-linguistic generalisations. This was achieved by eliminating the construction-specific transformational rules of the earlier theory in favour of a single general transformational rule  $\text{move-}\alpha$  (move anything anywhere), which mapped from trees at Deep-Structure

(DS) generated by abstract Xbar Theoretic phrase structure rules to those at Surface-Structure (SS), and then on to Logical Form (LF), the representation which interfaces with the conceptual-intensional (C-I) cognitive systems, and Phonological Form (PF), the representation which interfaces with the articulatory-perceptual (A-P) systems. This basic T-model (which is also sometimes referred to as the Y-model) architecture is shown in fig 2.1.

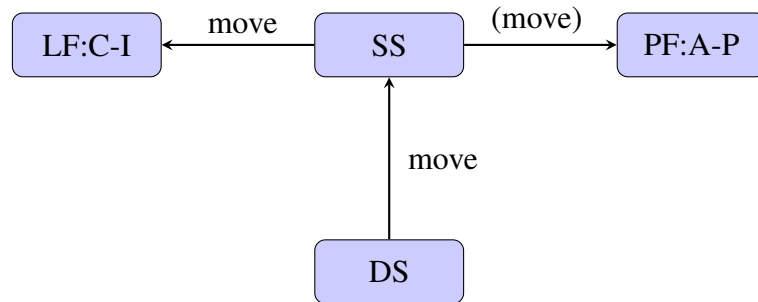


Figure 2.1: GB T-model architecture. Transformational movement rules applied overtly between DS and SS and covertly (i.e. with semantic effects but no visible phonetic effects) between SS and LF. Some authors proposed that in addition to such LF-movement, languages also exhibit PF-movement, which has visible phonetic effects but no semantic effects; an example of this is the rightward movement operation that was at one time a popular choice for describing phenomena such as heavy NP shift and extraposition.

Move- $\alpha$  was an extremely powerful rule, but its application was constrained in GB theory by various constraints or so-called *Principles*. Some of these Principles restricted the application of Move- $\alpha$ ; the Empty Category Principle (ECP), for instance, was argued to provide an account of some of the locality constraints which had recently been identified (Ross, 1967; Chomsky, 1973; Chomsky and Lasnik, 1977). Other Principles were argued to force movement to occur. One such Principle was the Case Theoretic requirement that all DPs be assigned structural case at SS, which often necessitated that they move overtly to a case-assigning position (this was referred to as the *Case Filter*). This enabled a unified analysis of passive, unaccusative and raising constructions, for instance, all of which were argued to involve verbs which were incapable of assigning accusative case to an object which was therefore forced to raise to the subject position. The idea was then that all humans are born with innate implicit knowledge of the Principles of the language faculty, thus accounting for the commonalities that show up across different languages. At least some of these Principles were assumed to be *parameterised*, however, with different parametric settings yielding dif-



ferent surface effects, thereby also accounting for cross-linguistic variation. The idea was that the child need only learn the idiosyncratic settings of a relatively small number of (possibly binary) parameters for their particular language, thus greatly simplifying the acquisition task. This came to be known as the *Principles and Parameters* approach to linguistic theory.

## 2.2 Minimalism

GB unearthed many important generalisations and constraints, and the Principles and Parameters approach was considered by Chomskyans to have provided a reasonable though by no means fully worked out approach to Plato's Problem. One concern with the theory which began to emerge, however, was that it had grown very complex, attributing to UG an arguably implausible amount of innate internal structure. In the mid 1990s, Chomsky (1993, 1995) therefore initiated his Minimalist Program (MP) for linguistic theory. MP asks to what extent language can be considered a 'perfect' system in the sense that its properties are determined solely by legibility conditions placed upon it by external systems of the mind/brain - in particular, the C-I and A-P systems - and by 'general considerations of conceptual naturalness' such as 'simplicity, economy, symmetry, nonredundancy' etc (Chomsky, 1995, page 1). This is referred to in Chomsky (2000) as the Strong Minimalist Thesis.

Whereas GB attributed to UG a rich, modular structure, Minimalism aims to minimise its content as much as possible by showing that many of the descriptive generalisations previously attributed to its various modules can actually be explained by the requirements of other cognitive systems, and by general principles of computational efficiency (what Chomsky (2005) refers to as 'third factor' principles). As an example of the latter, many of the locality effects enforced in GB by the ECP were reinterpreted in Minimalism in terms of general cognitive principles of least effort, such as the need for movement steps to be as short as possible. As an example of the former, take GB's Case Filter, which stipulated that all DPs be case-marked at the linguistic-internal level SS. In Chomsky (1995) SS is eliminated and lexical items already enter the derivation fully specified for properties such as case; the Case Filter is then reinterpreted as the need for these case features to be checked and deleted before the derivation reaches the LF:C-I interface, the rationale being that such features are semantically (though not phonetically) uninterpretable. Checking relations between a licensee (say, a DP with case) and a licensor (say a transitive verb) are established by moving the licensee (ei-

ther overtly or covertly) to become a specifier of the licensor.

A primary motivation for the initiation of MP was the desire to provide an account of human linguistic competence which is plausible from an evolutionary perspective (see Hauser et al. (2002) and also Hornstein (2009) for discussion): given that language is commonly supposed to have arisen around 50,000-100,000 years ago, it is argued, this does not seem to have been enough time for a rich, modular, innate linguistic system to have evolved. UG is therefore likely to be the result of just one or two mutations combined with other pre-existing and more general cognitive capacities.

MP therefore aims to dispense with the idea of grammar-internal constraints and levels of representation, eliminating both DS and SS (and, with the introduction of phases in Chomsky (2000), even LF (see Chomsky (2001, page 15; 2004, page 151)). It also aims to provide a much more explicit and operationally-based account of the generative procedure itself. Syntactic derivations now begin from a numeration  $N$  of lexical items  $\{A_i, B_j, C_k\}$  (where the subscripts indicate the number of occurrences of that item in the sentence) and are built up in step-wise fashion by selecting items from the numeration and then applying Merge and Move operations on them and the constituents derived from them. Merge and Move replace Xbar Theory and Move- $\alpha$  respectively, with Move (and for some authors also Merge) being driven by the need to check and delete features. The differential distribution of licensor features on (functional) lexical items across different languages now standardly provides the parametric account of language variation (the so-called *Borer-Chomsky Conjecture* (Borer, 1984; Chomsky, 2001b)).

Whereas in GB, Xbar rules applied strictly before Move- $\alpha$  rules, in MP, Merge and Move rules intersperse freely, generating so-called *bare phrase structures*. At certain points Spell-Out operations apply to split the derivation into two parts, leading to LF/C-I and PF/A-P representations respectively. In early Minimalism, Spell-Out applied only once per derivation, so that the basic architecture remained a T-model one, with the LF, and for some authors also PF, movement cycles retained, as shown in fig 2.2. With the introduction of phases in Chomsky (2000), however, Spell-Out now occurs at multiple points in the derivation, with small chunks of structure being transferred directly to the C-I (and presumably A-P) systems, thereby eliminating the LF (and PF) movement cycle(s), as shown in figure 2.3.

As will be clear from the preceding discussion, Minimalism can be divided into two quite distinct periods. Early Minimalism is defined by Chomsky (1993, 1995) before the introduction of phases and the probe-goal/Agree framework in Chomsky

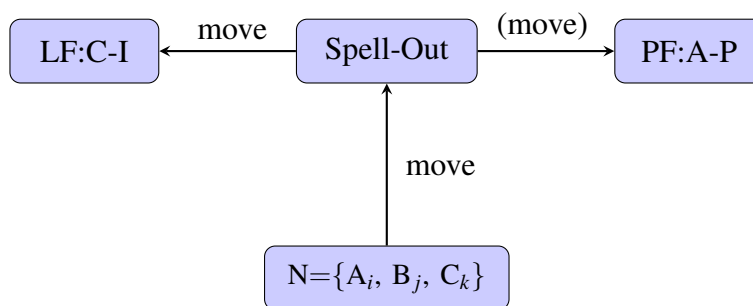


Figure 2.2: Early MP T-model architecture.

(2000, 2001b). The long-distance operation Agree of the later theory replaces many instances of covert movement in the earlier one, and whereas checking of uninterpretable features in earlier Minimalism necessitated (overt or covert) movement of the licensee to become a specifier of the licensing head, the long-distance operation Agree can apply with or without concomitant movement of the licensee. Uninterpretable features no longer enter the derivation fully specified, but must now acquire a valuation from interpretable counterparts before being deleted. Agree is driven by the need for verbal heads to have their uninterpretable phi-features (i.e. person, number and gender) valued and deleted by the corresponding interpretable features on nominals, and for nominals to have their uninterpretable case features also valued and deleted by verbal heads. Case features are thus now at least partially dissociated from movement, which is driven by the grammar-internal EPP feature, which is a nod to the Extended Projection Principle of GB.<sup>1</sup>

## 2.3 Minimalist Grammars

*Minimalist Grammars* (MG) is the name used to describe the formalism first introduced in Stabler (1997) and since developed further by Stabler and others. MGs are a rigorously formal, computationally oriented and polynomially parseable interpretation of many aspects of MP. Like MP grammars, MGs can be used to generate bare phrase structures, and include phonetically null heads and movement operations driven by the need to check and delete features, and constrained by a strict version of Chomsky's (1993) Shortest Move Constraint, and sometimes by other constraints on movement

<sup>1</sup>It should be noted that there is considerable variation between authors in the MP literature over many of these points. Epstein and Seely (2006), for example, argue that the EPP feature should be eliminated, with phi-feature/case checking/valuation driving movement.

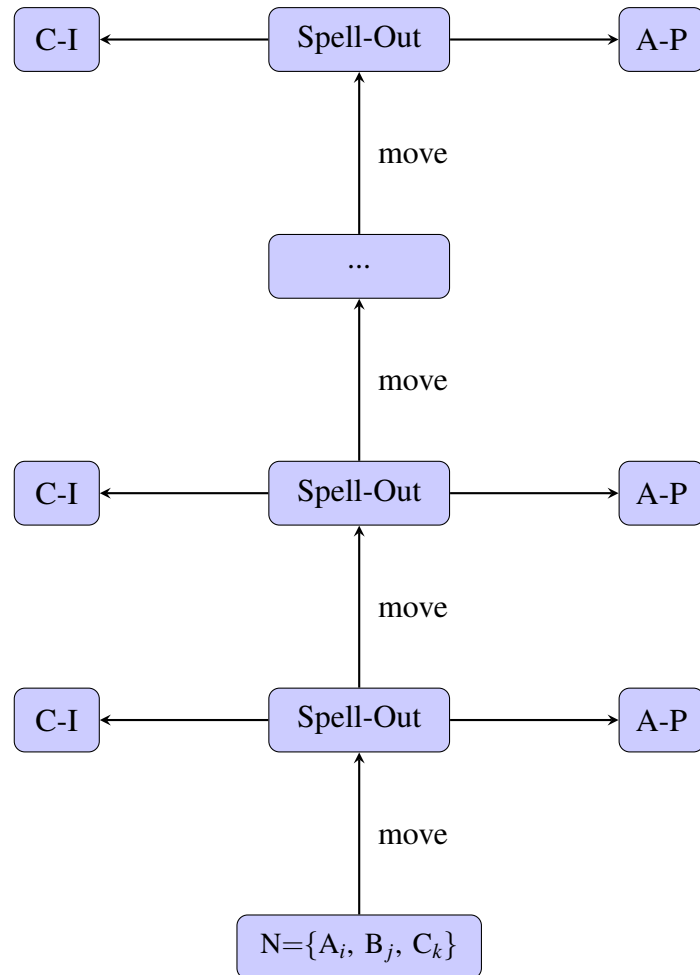


Figure 2.3: Late MP multiple Spell-Out architecture.

proposed in the TG literature such as the Specifier Island Constraint (Stabler, 2013b). MGs as usually defined are arguably most similar to early MP. For example, licensee features are always checked in a spec-head configuration, case features still drive (overt and covert) movement operations, there are no long-distance Agree operations (depending on how we interpret the covert movement operations included in some MGs), and no phases, at least as these are defined by Chomsky.

There are a number of notable differences between Stablerian MGs and (early) MP, however. Some of these are arguably due to the fact that MP is intended as a model of linguistic competence, whereas MGs are oriented more towards modelling grammars as they are utilised by the performance systems, in particular by the parser. As discussed by Stabler (1984), the relation between theories of grammar and parsing systems is not obvious, and we should resist the temptation to view parsers as direct or full implementations of competence theories: competence theories often aim for a higher degree of abstraction and succinctness than may be required for parsing, because they are aimed primarily at explaining language acquisition.

One significant difference between early MP and MGs is that where the latter include covert movement operations (e.g. Stabler (2001b); Harkema (2001)), this is not implemented as a separate covert movement cycle occurring post-Spell-Out. Instead, overt and covert movement operations are allowed to intersperse freely (just as Move and Agree operations do in later Minimalism), enabling derivations to remain strictly monotonic (i.e. every operation extends the root of the tree in accordance with the Extension Condition of Chomsky (1993)), and deriving a single phrase structural representation encoding both phonetic and semantic information (this is done by assuming that covert movement is simply movement of formal features but not phonetic ones, in a similar vein to the Move-F approach proposed in chapter 4 of Chomsky (1995)). There is also no appeal to a numeration to explain descriptive generalisations.

There are other differences between MGs and all iterations of MP which quite clearly arise from the need for MGs to be more precisely specified so as to be (efficiently) implementable on computers. One of these is that features on MG categories are strictly ordered, whereas this issue is usually left unspecified in mainstream Minimalist analyses. Furthermore, in MP, only licensor and licensee features (i.e. those driving Move) are usually discussed explicitly; a tacit assumption by many seems to be that while the application of Merge is constrained by a word's subcategorisation frame, this operation is not driven by the need to check/value/delete features. Chomsky (2008) does note that an expression must have a property or feature allowing it to

be Externally Merged, which he refers to as an *edge feature*, but does not develop a precise account of what these features are or what happens to them when Merge applies. As Chomsky observes a few paragraphs later “it has always been presupposed that E[xternal] M[erge] comes for free: no one has postulated an EPP property for EM,” which is tantamount to saying that External Merge is not feature driven in the same way that Internal Merge (i.e. Move) is.

Another difference between the two frameworks is that MGs generally include more operations than are generally assumed in Minimalism. For example, instead of a single operation Move (which Chomsky has even attempted to suggest is really just (Internal) Merge), separate rules are required in MGs to capture standard overt phrasal movement, head movement, covert phrasal movement, across-the-board movement and so on, and also to capture whether or not items do or do not move (further) following an application of a Merge or Move rule; however, in standard MGs, the union of the individual Merge rules can be viewed as a single function (and similarly for Move) so it is arguably possible to maintain the notion of a single Merge rule and a single Move rule, though collapsing Merge and Move in the way Chomsky suggests is not possible in MGs. The specific MG to be proposed in this dissertation extends the basic formalism considerably, including around 50 rules in total. As will be noted in what follows, at present some of the Merge rules operate over the same domain (and similarly for the Move rules), meaning that their union is a relation rather than a function; this will be rectified in a future iteration of the formalism.

Since the introduction of the *collapsed tree*, or *chain-based* approach, in Stabler (2001b) and Harkema (2001), MGs have standardly also differed from mainstream Minimalism in featuring essentially type-driven rather than structure-driven derivations, and in being viewed as generators of derivation trees rather than phrase structure trees. The next chapter includes a detailed discussion of these aspects of the formalism. For now, we will simply note that under the chain-based/type-driven perspective, the majority of the hierarchical relations between constituents are not accessible by the computational system as the derivation proceeds (though the phrase structure tree can subsequently be retrieved deterministically from the derivation tree). This simplification affords MGs some important computational advantages. For example, the set of MG derivation trees is a regular set, with string languages at their frontiers which are context free. This is one reason why well-studied context free parsing techniques, such as CKY and Earley parsers, have been successfully adapted to MGs.

The formal precision of the MG formalism has enabled a number of precise in-

vestigations into the expressive power of these grammars. For example, it is known that standard MGs (Stabler, 1997) are weakly MCFG-equivalent (Michaelis, 1998; Harkema, 2001) and therefore polynomially parseable, that their additional power over and above context free grammars derives from their ability to generate remnant movement<sup>2</sup> structures (Kobele, 2010), that adding a mechanism allowing for percolation of licensee features takes them out of the class of context sensitive grammars (Kobele, 2005), that without Stabler’s strict version of the Shortest Move Constraint, the membership problem of MGs is as difficult as provability in Multiplicative Exponential Linear Logic (Salvati, 2011), and that adding an additional Specifier Island Constraint (SpIC) makes them mildly context sensitive in Joshi’s (1985) sense (Kanazawa et al., 2011), though still more powerful than the near-context free TAG class grammars. Although MGs are polynomially parseable, their worst-case complexities are high order polynomials. More specifically, MGs without head movement can be parsed in  $O(n^{2k+3})$  time (Fowlie and Koller, 2017), while those with head movement can be parsed in  $O(n^{2k+5})$  time (Stanojević, 2019), where  $k$  is size of the set of movement licensee features in the grammar.

Finally, MGs include a much stricter version of the Shortest Move Constraint than Chomsky’s (1993) original formulation, which held that in cases where there are two potential movers to a given target position, it is the mover that is closest to that target position which must move, explaining contrasts such as the following.

- (4) a. Who<sub>*i*</sub> did Jack persuade *t<sub>*i*</sub>* to do what?  
 b. \*What<sub>*i*</sub> did Jack persuade who to do *t<sub>*i*</sub>*?

Chomsky’s SMC (and the later Minimal search approach to intervention effects of the probe-goal framework) is defined over phrase structural configurations. However, as noted above, standard MG derivations disregard the internal phrase structural geometry of the tree, meaning that there will be no way for the system to know that *who* is structurally superior to *what* in 4. Chomsky’s version of SMC is therefore simply unformulable here. Instead, Stabler introduces a much stricter version of the SMC which simply bans all derivations in which two movers have the same first feature. Salvati (2011) shows that Stabler’s SMC is required in order to keep MGs weakly equivalent to MCFGs. However, while this strict version of SMC can successfully enforce a number of well-known locality effects, such as *wh*-islands (5b) and the ban on superraising

---

<sup>2</sup>Remnant movement refers to the movement of a constituent which has itself already had one of its subconstituents moved out of it. See Stabler (1999) for discussion of this type of movement in an MG context. The MG presented in this dissertation makes occasional use of remnant movement.

(6c), for instance, it can also be overly restrictive in some instances. For example, without adopting additional mechanisms, and assuming we wish to use only a single type of -wh licensee, then as well as correctly blocking the derivation of 4b, Stabler's SMC will also incorrectly block 4a.

- (5) a. What did Jack think that Pete said?
- b. \*What did Jack wonder who said?
  
- (6) a. It seems that Jack is believed to have helped.
- b. Jack seems to be believed to have helped.
- c. \*Jack seems that it is believed to have helped.

## 2.4 Parsing with Transformational Grammar

Beginning in the mid-1960s, there have been a number of parsing systems built which can be viewed as implementing aspects of the various iterations of Chomskyan syntactic theory, and several of these systems are reviewed in this section. With the exception of Dekang Lin's Principar and Minipar parsers, none of these parsers have had wide-coverage by modern standards.

### 2.4.1 Early TG parsing

There were some early attempts to implement the (Extended) Standard Theory of Chomsky (1957, 1965) in parsing systems (e.g. Petrick (1965) and Zwicky et al. (1965)), but these systems ran into some severe efficiency issues which meant that these early TG parsers showed only moderate successes in highly restricted domains. One fundamental issue was that, as shown by Peters and Ritchie (1973), the automata theoretic complexity of early TG was overly powerful (Turing Complete).

A further issue was that, when implemented faithfully, the general architecture of the (Extended) Standard Theory was not very well suited to the parsing task. Recall that this model consisted of two components, a context-free base generating deep structures and a transformational component mapping these to surface structures. This works well enough for generation, but for parsing it must be applied in reverse. This is precisely what these early TG parsers attempted to do. First, a set of candidate surface trees was generated using a context free grammar, after which a set of transformations were applied in reverse to produce a set of candidate base trees. Next, a check was



performed to see which of these candidate trees were in fact generable by the base, after which the transformations were applied again, this time forwards, to check that each surface tree was actually derivable from its base tree. According to King (1983, page 28), this last step was necessary because it was possible to arrive at legitimate base trees from illegitimate surface trees owing to the fact that reverse transformations could not always be constrained in the same way as their forward counterparts.

Unsurprisingly, this complex procedure led to a host of efficiency problems. For example, because the candidate surface trees were generated by a context free grammar, and because such a grammar is less powerful than a TG grammar, a whole host of spurious surface trees would be presented as initial candidates, and these then had to be filtered out by the reverse transformations. A further issue was that the transformations could in principle have been applied in any order in their forward applications and so this led to an explosion in the search space of the parser. Finally, transformations which deleted material were especially difficult to reconstruct in reverse, since there was often no way to know what had been deleted from where.

## **2.4.2 Augmented Transition Networks**

Augmented Transition Networks (ATNs) are introduced in Woods (1970, 1973) as an extension to the simpler Recurrent Transition Networks (RTNs). RTNs are essentially a graphical implementation of a push down automaton or a set of context free phrase structure rules. ATNs extend RTNs by incorporating special registers in order to represent more powerful rule systems. For instance, context sensitive rules require that the system can take into consideration what has gone before and what is yet to come before deciding to follow a particular transition. This requires additional memory beyond the simple stack of the PDA, so that information can be stored and retrieved on demand. Upon encountering a *wh*-constituent, for instance, the ATN parser would place this element into a hold register, from where it could later be retrieved if the parser encountered a gap.

The ATN was in fact a very efficient parser for its time, although it was only ever used in closed domains, such as in a question-answering system for moon rocks. The secret behind its success was probably due at least in part to the fact that, unlike the earlier TG parsers, it did not attempt to construct both deep and surface structure. Instead, it built only deep structure, which is the level at which all thematic relations are encoded. This could be seen by some as making the ATN less faithful to TG theory

than those earlier parsers. However, as noted above and discussed in Stabler (1984), it is not necessarily the case that the human parser computes all levels of representation that are specified by the theory of competence.

It is, however, also worth noting that an approach to antecedent-trace recovery that relies on registers will only work for the canonical case in which the antecedent precedes its trace. However, contemporary Minimalist theory allows for so-called remnant movement, which results in traces preceding their antecedents. ATN parsing would therefore not be a suitable framework for implementing modern TG.

### 2.4.3 Parsifal

Perhaps in reaction to the problems of indeterminacy faced by early TG parsers, Marcus (1980) proposed that human languages can in fact be parsed deterministically. His Parsifal parser proceeded from left to right and used an active node stack holding constituents currently under construction, as well as a limited-size look-ahead buffer containing unparsed words and constituents not yet integrated into the structure. Movement transformations were not simulated directly as in the earlier TG parsers, but instead trace nodes were inserted and linked back to NPs encountered earlier in the sentence. In contrast to the ATN which constructed only deep structure, this parser constructed only surface structures, though these did include traces of movement for recovering predicate argument relations. All structures built during the course of parsing were used in the final structure with no backtracking at all. Marcus argued this deterministic strategy could explain the existence of garden path sentences where the human parser apparently commits to a single ultimately incorrect analysis.

Marcus' parser is criticised in Joshi (1989) for being difficult to formalise owing to the fact that the rules of the grammar and the mechanisms of the parser are not sufficiently distinguished. As such, it is not possible to state precisely all the analyses which are and are not supported by this system. Ultimately, of course, deterministic parsing was overshadowed by the success of inherently nondeterministic, statistical parsing techniques.

### 2.4.4 Principles-based parsers

A number of parsers emerged from the tradition of the Principles-based GB framework (e.g. Kuhns (1990), Dorr (1991), Johnson (1991b), Fong (1991)). However, while GB's more constrained formalism rescued these parsers from some of the problems

faced by the early TG parsers, its generate-and-filter paradigm introduced a new set of problems. Many of these systems worked by first using Xbar rules to generate a set of candidate surface trees and then applying the various Principles to those trees in order to filter out illicit structures. This led to initial massive overgeneration with a great many trees needing to be traversed and checked for various constraint violations. Furthermore, although there was now no need to process transformations in reverse, the complex interactions of the constraints, whose relative ordering was left unspecified by the theory, also caused various efficiency issues. In almost all cases these parsers were not wide-coverage by modern standards, being equipped with relatively small domain-specific lexicons.

### 2.4.5 Principar

One GB parser which deserves a special mention is Lin's (1993) Principar parser, which (along with its successor Minipar) is to my knowledge the only truly wide-coverage parser to have emerged from the Chomskyan tradition besides that presented in this dissertation. Principar had a lexicon of over 90,000 entries, and was also more efficient than the other GB parsers discussed above, which is no doubt what enabled it to be applied to the computationally intensive task of wide-coverage parsing.

Principar's grammar and parser are precompiled into a network of nodes such as VP, Vbar, V, NP, Nbar, N, TP, Tbar, etc, representing allowable dominance and subsumption relations. Parsing consists of passing structural description items representing constituents up through this network from its lexical leaf nodes to its CP root node. As these items reach the nodes in the network, the system attempts to combine (i.e. Merge) them, performing various unification checks and also applying various Principles in order to ensure that the combination is licit. Each item contains a reference to the span it covers in the string, pointers to the items that derived it, and an attribute-value vector representing all and only the information that is required for processing that constituent. For instance, there are features indicating whether or not a constituent must be case marked, whether it contains a moving element, whether it has already passed one barrier (see Chomsky (1986a) on the concept of a *barrier*) and hence may not pass another etc. The Principles of the grammar are thus applied to these structural descriptions as the trees are constructed, allowing the derivations to be aborted as soon as the offending structure is encountered, with no need for performing the computationally costly tree traversals required by other Principles-based parsers. In

fact, because the structural descriptions are attribute-value matrices of a constant size, checking that a structural description satisfies a Principle is guaranteed to take constant time.

It should be noted, however, that although Principar was undoubtedly a Principles-based parser, it is not clear that it could accurately be described as Transformational. Movement is achieved by passing features up the tree from the site of the trace to the site of the antecedent. The antecedent is then merged directly into its surface structure position, with the constraint that it must be unifiable with the features of the mover. As Lin himself notes, this approach merely simulates Move- $\alpha$  rather than modelling it directly, something which Lin states would lead to far too much overgeneration. This approach to movement is therefore more akin to the slash-feature passing mechanism of the non-transformational GPSG and HPSG formalisms than what is generally assumed in TG.

## 2.4.6 Minimalist parsers

There have been a number of parsing systems constructed within a Minimalist framework. In this section we review several of these.

### 2.4.6.1 Minipar

Minipar (Lin, 1998, 2001) is the Minimalist descendent of the Principar parser discussed in the previous section. There is far less documentation on the inner workings of Minipar, but what is clear is that this is essentially an updated version of Principar that incorporates MP's bare phrase structure and some of its economy principles. It also has an expanded lexicon of about 130,000 entries derived from WordNet (Miller et al., 1990) and incorporates a statistical model, having been self-trained on a 1GB corpus. Although in its inner workings Minipar manipulates constituency trees, it outputs dependency trees using a generative model to calculate the probability of each tree as a product of the probabilities of all the dependencies it contains. Like Principar, Minipar is very efficient, able to parse 500 words per second on a Pentium-III 700Mhz with 500MB memory (Lin, 2001, page 2), and achieving 79% recall and 89% precision in recovery of dependencies in the SUSANNE corpus.

With so little documentation on Minipar, it is difficult to assess the formal properties of this parser, or to know which structures its grammar can and cannot generate. Presumably, its treatment of movement is similar to the non-transformational GPSG-

style approach adopted by Principar. One issue with this approach is that it relies on traces always being c-commanded by their antecedents (Lin, 1993, page 115), making it unsuitable for implementing many contemporary Minimalist proposals involving remnant movement.

#### 2.4.6.2 MG Parsers

A number of working parsers have also been developed for Stablerian MGs, which have been very rigorously defined and studied, and which do allow for remnant movement. Harkema (2001) introduces variants of the context-free CKY and Earley algorithms for MGs, using a collapsed-tree notation which essentially treats MG constituents as categories, rather than as structures. These parsers do not build phrase structure trees directly, but instead build derivation trees whose yield languages are context free and therefore amenable to context free parsing techniques. Although phrase structure trees can be recovered deterministically from the derivation tree, many (perhaps most) MG researchers have now come to view the derivation tree itself as the primary syntactic data structure. This point is made explicit in Graf et al. (2017, page 8).

Since Harkema's seminal work, there have been various other MG parsers developed. Hale's (2003) implementation and is essentially the bottom-up CKY recognizer from Harkema (2001) augmented with a simple PCFG-style generative model whose probabilities were extracted from a corpus study by Keenan and Hawkins (1987). Mainguy (2010) presents a probabilistic top-down MG parser, which works by first converting the MG into a strongly equivalent LCFRS before employing standard top-down LCFRS parsing techniques. Stabler (2013) presents a top-down incremental recursive descent-style parser with a beam search mechanism, which works by traversing feature structures in a graphical representation of the lexicon, unchecking features as it goes and building an MG derivation tree in reverse. There have also recently been proposals for transition-based Stanojević (2016) and left-corner MG parsers (Hunter, 2018; Stanojević and Stabler, 2018).

What all working MG parsers have until now shared in common is that they have been small-scale theoretical endeavours, essentially proofs of concept equipped with only toy grammars and lexicons; there has so far been no attempt to build MG parsers capable of parsing arbitrary sentences from the Wall Street Journal section of the Penn Treebank. There has been a limited amount of research into probabilistic MG parsing, most notably in generative locally normalised models (Hale, 2003; Hunter and Dyer,

2013). However, these works remain so far untested owing to the unavailability, until now, of any MG treebank for training and testing models.

The largest Minimalist Grammar to date of which I am aware is that which was created by Sarah Van Wagenen building from the base grammar for sentences containing relative clauses presented in Hale (2003). Van Wagenen's grammar contained 561 MG categories which were used to hand annotate 167 short sentences (in many cases individual clauses extracted from larger sentences) from the first chapter of Lewis Carroll's *Alice in Wonderland*. This resulted in a corpus of 2,150 word tokens which was also to my knowledge the largest MG corpus to date; the lexicon extracted from this corpus contains 1,351 (overt and null) entries. By comparison, the grammar to be presented in this dissertation contains 1088 MG categories and has been used to hand-annotate 1036 sentences, the majority of which come from the Penn Treebank and range between 5 and 25 words in length. The automatic annotator was then trained on this initial seed set and used to generate trees for a further 21,000 sentences of between 1 and 30 words in length. The entire MGbank contains 317,647 word tokens and the lexicon extracted from it consists of 40,182 (overt and null) entries.

It is also worth noting that Van Wagenen's grammar was also not, strictly speaking, a single grammar, but rather 167 separate grammar fragments, each designed to cover one sentence in the text. The separate grammars contained many categories in common, of course, but there was no machine testing carried out to check the global consistency of the grammar and the extent to which it over- and undergenerated. The larger a grammar becomes, the greater its tendency to overgenerate in unexpected ways, which as well as leading to spurious analyses can also making parsing very inefficient. Of course, Van Wagenen's grammar was never intended to be applied to the wide-coverage parsing task, but was instead used to carry out psycholinguistic experiments intended to evaluate the extent to which readers' fMRI data correlates with abstract Minimalist proposals for syntactic structure (Brennan et al., 2016). In this context, the extent to which the grammar overgenerated was not a primary concern. That said, the availability of a globally consistent and more realistic wide-coverage MG parser should open up new avenues for this kind of psycholinguistic research.

### **2.4.7 The Minimalist Machine**

For a number of years, Sandiway Fong and Jason Ginsburg have been developing a system which they call the 'Minimalist Machine' (see, e.g., Fong and Ginsburg (2012)).

Their project website describes this as ‘a computer implementation of a theory within the Minimalist Program’ with the goals of ‘assembling a substantial, consistent and coherent theory based on the probe-goal framework in the Minimalist Program,’ and ‘demonstrating derivations for a wide variety of examples.’

The Minimalist Machine differs in a number of important ways from the work presented in this thesis. First, it is not currently implemented as a parser. Rather, it is a derivation simulator which starts from a numeration which is prespecified by the linguist for a given sentence and which contains all and only the correct lexical items required to derive that sentence. At a conference on Minimalist parsing at MIT in 2016, Fong and Ginsburg presented this system as a tool which linguists could use to check the correctness of their analyses. The second major difference is that the Minimalist Machine is a theoretical, competence-oriented computational system which aims to very faithfully implement many of the most recent proposals in the Minimalist literature. Although Fong and Ginsburg’s system has been applied to a few hundred different constructions, they have not attempted to handle the messy data found in realistic corpora, nor to construct a wide-coverage lexicon or corpus. Instead, they have focused on specific example sentences found in linguistics textbooks and other publications, rather than on the data found in corpora such as the Penn Treebank. The present work, on the other hand, attempts to strike more of a balance between competence-based and performative computational linguistics, but is as a consequence less faithful to mainstream Minimalist theory.

## **Chapter 3**

# **Extended Directional Minimalist Grammars**



### 3.1 Introduction

The wide coverage MG constructed for the current project uses an extended version of Stabler’s original formalism that incorporates a number of proposals that were subsequently put forward by other MG researchers; it also introduces several novel mechanisms into the framework. We will refer to the resulting formalism as *Extended Directional Minimalist Grammars* (EDMGs). All of these extensions are designed to enhance the linguistic sophistication of the formalism, enabling it to more adequately describe a considerable range of construction types, while also avoiding overgeneration as far as possible.

This chapter presents the EDMG formalism in its entirety, including detailed descriptions of some of the constructions in MGBank. A wider sample of the grammar can be found in Appendix A, which lists a selection of the lexical entries from the treebank, in many cases accompanied by notes of the construction in question and example sentences.

### 3.2 Directional Minimalist Grammars

MG is a strongly lexicalised formalism, which means that the majority of syntactic information resides in the lexicon and there are a relatively very small number of (around 50) abstract, cross-categorical Merge and Move rules. The basic building blocks of the MG presented in this dissertation are words, which are treated here as unanalysed units. Note that this is not the case either in mainstream Minimalism or in theoretically oriented work on MGs, where for instance tense and number inflections start out as separate syntactic heads that end up affixed to their stems either via affix hopping or head movement operations. The reason this approach was not pursued here is that during parsing words are encountered as unanalysed units and so separating affixes from their stems would have entailed a separate morphological parsing step that would invariably have introduced errors into the pipeline.<sup>1</sup>

MG lexical items consist of sequences of three basic and disjoint types of features: phonological, semantic and syntactic.<sup>2</sup> The phonological features will here simply

---

<sup>1</sup>It has in fact been shown that separating affixes from their stems during parsing does not improve performance, despite the common sense intuition that this should help the parser to generalise better to unseen word forms (Manning and Schütze, 1999, pages 131-133).

<sup>2</sup>Stabler (1998, page 85) points out that this fact apparently differentiates MG from MP, which often confounds the distinction between syntactic, semantic and phonological features. For example, Chomsky designates certain syntactic features as ‘interpretable’ (such as person/number on DPs). Others,

be represented by the word in italics, for example *the* will be the representation of the phonological features of the definite article. This dissertation will not address the semantic representations which are derived by MGs. These features will therefore be omitted entirely, although I assume that in operation behind the scenes is something along the lines of Kobele's (2006) direct compositional approach to MG semantics, which associates a semantic value with each syntactic feature, and semantic operations such as storage and retrieval and functional application with the syntactic operations Merge and Move (see section 3.2.5 for further discussion).

MG derivations are constructed bottom-up using two basic structure building operations: Merge and Move. These operations can be formalised as deductive rules and are composed of separate sub-rules for handling different scenarios (e.g. for the case where a constituent begins moving after being merged as the complement of some head vs. the case where it does not move). Merge combines two syntactic objects which are external to one another, and is thus unequivocally binary in nature. Move also takes two arguments, a main tree  $T$  and a subtree  $t$  of  $T$ , and merges  $t$  with the root of  $T$ . For this reason, Chomsky often refers to Move as *Internal Merge*, in contrast to standard *External Merge*. However, owing to Stabler's strict version of the Shortest Move Constraint (discussed below), the choice of subtree  $t$  is in an MG always deterministic, hence it is only necessary to specify  $T$  as the sole argument of Move. As we shall see, this means that while Move generates binary branching derived phrase structures, it generates unary branching derivation structures, and this fact turns out to be the key to the polynomial parseability of MGs, because it means that the strings at the fringes of MG derivation trees are context free.

MG lexical items contain sequences of syntactic features ordered from left to right, and it is these features which drive the derivation by determining the application of Merge and Move operations. Both Merge and Move match, or *check*, and (usually) delete features, and a convergent derivation is one in which all syntactic features have been deleted with the exception of a single  $c$  feature on the complementizer head of the matrix clause (equivalent to reaching the root  $S$  or  $Sbar$  node in a traditional phrase

---

meanwhile, are designated 'uninterpretable', but may have phonological effects (such as case on DPs). However, if we adopt the direct compositional approach of Kobele (2006), which associates a semantic feature, or value, with each syntactic feature, it is easy to see how Chomsky's viewpoint can be made to fit with MGs. Under this approach, interpretable syntactic features would be those associated with some meaningful semantic value, while uninterpretable ones would simply be associated with the identity function. The morphophonological effects of certain features, meanwhile, can be simulated in MGs under an approach which separates inflections such as case markings from their stems and associates the corresponding morphosyntactic features with these inflectional morphemes. As noted above, however, we do not implement morphological processes in the grammar presented in this dissertation.

structure grammar). This conception of feature checking differs somewhat from what is standardly assumed, at least explicitly, in MP, where it is generally only Move (and in later theory Agree) which is seen as being driven by the need to check (morphophonological) features, although there are exceptions to this (e.g. Collins (2002), (Chomsky, 2008, page 6), Adger (2010)). The fact that *devour* must take a direct object, whereas *eat* can be used intransitively, must of course be represented in the lexicon somehow. In MP the standard though implicit assumption seems to be that lexical items have a subcategorisation frame of some sort, and that Merge can apply freely as long as it conforms with this. MG instead encodes the subcategorisation frame directly into the same ordered feature sequence that contains the features driving movement. Given that in MP operations are assumed to be subject to a principle of Last Resort, according to which operations only apply if they must (in the case of Move, to check uninterpretable features), and given also Chomsky's more recent conception of Move as a species of (Internal) Merge, subjecting Merge to the same feature checking imperative as Move is an arguably very natural step.

MG syntactic features come in four types. *Selectee* features are similar to traditional part of speech categories and include n for nouns, v for verbs, p for prepositions, t for tense heads, c for complementizer heads etc. For each selectee feature  $x$ , there are corresponding selector features  $=x/x=$  which select for them. It is the selectee and selector features which drive (External) Merge. In the directional MG to be presented in this dissertation, the directionality of selection is indicated by placing the = diacritic to the right (for rightward selection) or left (for leftward selection) of the selectee symbol. Thus the MG feature sequence  $d= =d v$  is similar to the CCG category  $(S \backslash NP) / NP$  (abstracting away from the DP vs NP contrast), except that in MG it is not possible to define complex selector features; for example, the CCG category  $((S \backslash NP) / (S_{[to]} \backslash NP))$  used for raising and control verbs is undefinable in MG, because it involves the selection of an embedded clause that is specified as missing its subject.

In addition to selectors and selectees, there are also licenser +f and licensee -f features, which drive Move operations. A licenser feature on the head of some root projection will cause a constituent with a matching licensee feature to move and become a (specifier) dependent of that head, with the +f and -f features being deleted.

These four types of features are strictly ordered on lexical items so that any selectors and licensers precede any selectees and licensees, while the selectee feature (if present) must precede any licensees. In other words, taking *selic* to be the set of all selector and licenser features, *selectee* to be the set of all selectee features and *licensee*

to be the set of all licensee types, MG feature sequences (which we will often refer to as MG *types*) are all of the following form, where < means ‘precedes’.

$$(7) \quad \textit{selic}^* < \textit{selectee}^* < \textit{licensee}^*$$

These restrictions ensure that all constituents must be fully saturated (i.e. have all selectors and licensors checked and deleted) before they can themselves be selected for (encoding the Xbar Theoretic requirement that all dependents are maximal projections), and that only once selected for can they begin to undergo movement (i.e. once they have become a subtree of some larger tree). Note that the selectors and licensors may freely intersperse, so that a verb which attracts a mover may subsequently select for some other constituent and vice versa. Together, the selectors and licensors constitute the subcategorisation frame of a given lexical item.

### 3.2.1 A simple derivation using Merge

Consider the following possible types for the transitive verb *reads* and the pronouns *it* and *they*, which we will use to derive a simple VP for the transitive sentence *they read it* (:: is a type identifier indicating an underived syntactic object, i.e. a lexical item; it contrasts with : which will later be used to indicate derived items).

read :: d= =d v

it :: d

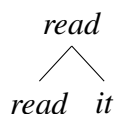
they :: d

In MGs, feature checking and deletion proceeds from the left to the right, one feature at a time.<sup>3</sup> The first feature of *read* is d=, indicating that this lexical item is looking for a DP (Determiner Phrase<sup>4</sup>) to its right. We will often refer to the first feature of any expression or moving subpart (chain) of an expression as the *active* feature of that expression or chain. Since *it* has an active d feature, we can merge these two items. In Chomsky (1995, page 243) the result of this merge would be the unordered set {*read*, {*read*, *it*}}, where the leftmost occurrence of *read* is a *label* indicating the head of the phrase. The head of a phrase determines the subsequent syntactic behaviour of that phrase; for example, this phrase will next need to be merged

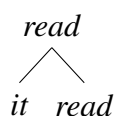
<sup>3</sup>Note that in MP the question of whether or not features are ordered is usually left unspecified.

<sup>4</sup>In Minimalism it is standardly assumed that pronouns are a type of intransitive determiner. Part of the motivation for this is the fact that many pronouns can be used pronominally as well as pronominally: *I read that (book) but haven't read those (books), we (Americans) are fond of you (Brits)*, etc.

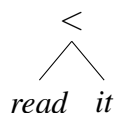
with a DP on its left owing to the =d feature that is now the leftmost feature of its verbal head. Chomsky (1995, page 246) also uses the following equivalent Bare Phrase Structure tree notation:



It is important to note that in Chomsky's set theoretic tree structure the linear precedence of terminal elements is not encoded directly by their left-to-right ordering (instead precedence is determined by a separate linearisation algorithm at PF along the lines of Kayne's (1994) Linear Correspondence Axiom (LCA)). The above tree is therefore, strictly speaking, entirely equivalent to the following one:

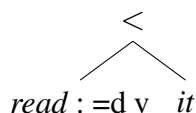


Stabler (1998, page 77) observes that Chomsky's notation is slightly confusing as it could be incorrectly interpreted as suggesting that the features of *read* appear twice in the tree. Stabler therefore suggests the alternative notation shown below, in which non-terminal labels point towards the head daughter.



This representation also contrasts with Chomsky's in that here linear order is intended to be represented directly in the left-to-right ordering of terminals on the tree's frontier.

To make things clearer, we will often also include the remaining syntactic features of the lexical items in the tree as follows:



The verb still has two remaining features, the active one being =d, indicating that this phrase now needs to merge with a DP on its left. The pronoun *it*, meanwhile, has had all of its features checked, and is therefore now syntactically inert.

We can now left merge the lexical entry for *they* with our newly derived phrase, resulting in the final structure shown in fig 3.1.<sup>5</sup> This type of representation is referred to as a *derived tree* in the MG literature, where it is the standard format for representing phrase structure.

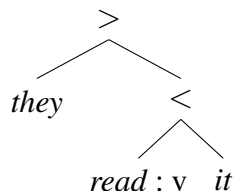


Figure 3.1: (Simplified) MG derived tree for the sentence *they read it*.)

### 3.2.2 Classical Xbar Theory, Derived Xbar Theory and MGs

In GB the theory of phrase structure was a version of Xbar Theory (Chomsky, 1970; Jackendoff, 1977) in which structures were at most binary branching (following influential proposals by Kayne (1984)). We will refer to this version of Xbar Theory as *classical Xbar Theory*. Chomsky (1995) argues that many of the core principles which had been stipulated in classical Xbar Theory can in fact be derived from more basic properties of the grammar. This led to the much more minimal, or *bare*, conception of phrase structure which we saw above. This section shows that Xbar Theoretic principles are also derivative notions within the MG framework, and introduces some useful terminology that will be used throughout this dissertation.<sup>6</sup> The derivation trees produced by the parser during the creation of MGbank were transduced into both MG

<sup>5</sup>The discussion makes a number of simplifications here; for example, it omits the null heads that would allow this structure to further project to the TP and CP levels (equivalent to the S and S' nodes in a traditional phrase structure grammar).

<sup>6</sup>Note that in recent work Chomsky 2013 (following up on ideas in Cann (1999)) has gone much further towards eliminating Xbar Theory entirely, suggesting that 'there is no concept SPEC, phrases need not be endocentric [i.e. headed], and projection (like order) is a distinct property'. The MGbank grammar does not follow these more recent ideas, instead adopting the classical version of BPS in Chomsky (1995). One reason for this is that these more recent ideas have yet to be fully worked out; another has to do with the shift in orientation within the MG formalism over the past ten years or so away from viewing phrase structure trees as substantive psychological constructs. Instead, the focus within MG is now on the derivation tree and the derived semantic logical form (see section 3.2.5). If phrase structure trees are, as most MG researchers now assume, simply artefacts of the way in which linguists choose to analyse sentences, then the question of how their nonterminals are labelled is no longer a coherent one. One could nevertheless ask a closely related question: why does the head determine the behaviour of a given syntactic object? In MG, as discussed below, this turns out to be an inevitable consequence of the strict ordering of the selector, licenser, selectee and licensee features, which can in turn arguably be derived from underlying semantic requirements.

derived trees and classical Xbar trees, and both of these tree types are included in the corpus. Classical Xbar trees were included because they are in many ways easier for humans to read precisely because they are not ‘Minimalist’ in the sense that they contain the bar level diacritics and co-indexed traces which Chomsky’s (1995) Inclusiveness Condition forbids.

In GB Deep Structures were created using abstract Xbar Theoretic PS rules, after which Surface Structures were derived by applications of move- $\alpha$ . The following are the two basic Xbar rules that were used for a head-initial language like English,<sup>7</sup> where X, Y and Z are abstractions over category types such as N, V, P, T, etc, and parentheses indicate an item which is in principle optional.

- (8) a.  $XP \rightarrow (YP) X'$   
 b.  $X' \rightarrow X (ZP)$

There are five theoretical claims encapsulated in the above rules, which are the following: 1. that all phrases have exactly one head (hence X alone appears on both the lefthand and righthand side of each rule); 2. Phrases are at most binary branching (but can be unary branching); 3. that all heads obligatorily project separate X' and XP projections, regardless of whether they have any dependents; 4. that all categories of head can in principle take up to one specifier (YP) dependent, defined as a sister to X' and daughter to XP, and one complement (ZP), defined as a sister to the head X; and 5. that dependents are always maximal projections (hence there are no rules like  $X' \rightarrow X (Z')$  or  $XP \rightarrow (Y) X'$ ).

Note that both the specifier YP and complement ZP are in principle optional, but whether or not they appear is (at least for complements) determined by an individual lexical item’s subcategorisation frame. Xbar Theory thus removed category-specific subcategorisation information from the rule system and in so doing eliminated a major redundancy from TG. This is because, prior to its introduction, such information had been stated both in category specific PS rules such as  $VP \rightarrow V NP$  and  $VP \rightarrow V$  (the rules for transitive intransitive verbs) and in the lexicon; for example, the fact that *devour* must take an object while *eat* can omit that object is an arbitrary fact of English and hence must be stated in the subcategorisation frames for these verbs.

<sup>7</sup>Additional rules such as  $X' \rightarrow X' WP$  and  $X' \rightarrow WP X'$  were also used to introduce rightward and leftward adjunct dependents (here, WP) as both sisters and daughters to X'. However, for reasons which are discussed in section 3.3.1, adjuncts must adjoin to XP in the MGBank Xbar trees, which is the standard assumption in Minimalism. The Xbar adjunction rules used for these trees are therefore  $XP \rightarrow XP WP$ , and  $XP \rightarrow WP XP$ .

In Chomsky's (1995) Bare Phrase Structure Theory, concepts such as head, specifier, complement and intermediate projection are shown to be derivable from more basic properties of the computational system (see Hornstein et al. (2005) for an in-depth discussion). This is also true within the MG formalism. Take the headedness (or *endocentricity*) principle, for example. In MGs, as in MP, phrases are not constructed as they were in GB using abstract Xbar Theoretic PS rules, but rather by interleaving bottom-up Merge and Move operations applied to lexical items and larger derived items. The constraints on the ordering of features shown in 7 above ensure that there can only ever be one lexical item  $L_h$  in any given structure which has unchecked selector or licenser features, because by the time an expression's selectee is exposed, its selector and licenser features will already have been deleted. Furthermore, (looking ahead slightly to section 3.2.4), any moving items inside the main structure with undeleted licensee features will only be accessible to a matching licenser feature on  $L_h$  (there is no *sideward movement* in the sense of Nunes (2004), although see section 3.3.7.2 on across-the-board movement). The system therefore has no choice but to continue working on working on  $L_h$  until such time as all of its selector and licenser features have been checked and its selectee feature is exposed at which point either it will be selected by a higher head, or if the selectee is a c feature, the derivation may be complete.  $L_h$  therefore fully determines the operations into which the root structure containing it can enter. In other words,  $L_h$  determines the *type* of the root structure, a property which linguists refer to as headedness.

The headedness principle of Xbar Theory thus falls out from the strict ordering of feature sequences on MG types (which in turn presumably derives from the requirements of the underlying combinatorial semantics). The binary branching requirement of Xbar Theory, meanwhile, is enforced in an MG, as in MP, by the fact that the Merge and (from a phrase structural perspective) Move rules take exactly two arguments. Chomsky (1995, page 226) argues that Merge is binary because it is the simplest operation which combines two syntactic objects to form a single object, such combination clearly being necessary in any linguistic system. From the perspective of a minimal and evolutionarily plausible theory of UG, a simple binary Merge operation is clearly more attractive than a more complex n-ary version (keeping structures strictly binary also severely restricts the number of possible bracketings which the language acquirer must entertain for each sentence they encounter). Notice too that the binary branching condition is now enforced even more strictly than in classical Xbar Theory, whose PS rules also allowed for unary branching structures in those cases where either the



complement or specifier was omitted, a point to which we return below.

Next, consider projection levels. In the transition from GB's Xbar Theory to Minimalism's Bare Phrase Structure Theory, the bar level diacritics ' and P used to distinguish minimal (X), intermediate (X') and maximal (XP) level projections were eliminated as substantive theoretical primitives in accordance with Chomsky's (1995, page 228) Inclusiveness Condition, which prohibits the output of the syntactic derivation from containing any elements which are not already contained on the lexicon. Bar level diacritics in Minimalism are therefore regarded as purely derivative notions, created only when binary Merge or Move operations apply, and definable solely in terms of the relations between different nodes in the tree, rather than as substantive features which the computational system can target. A minimal (X) projection is a terminal node; a maximal (XP) projection is a node which does not project any further, such as the root node and the terminal nodes *they* and *it* above; finally, an intermediate (X') projection is a node which is neither a minimal nor maximal projection, such as the node labeled < above.

Notice that on this relational and derivative definition of bar levels, the terminals *they* and *it* are simultaneously defined as minimal and maximal projections owing to the fact that they are lexical items which do not project further. This was not the case in classical Xbar Theory, where lexical items obligatorily projected all three levels of structure, regardless of whether they took any dependents. For example, we can use Xbar PS rules in 8 above to generate the classical Xbar tree shown in fig 3.2 for our earlier example sentence.

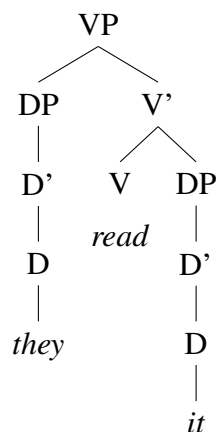


Figure 3.2: Classical Xbar Theoretic equivalent of the BPS tree in fig 3.1

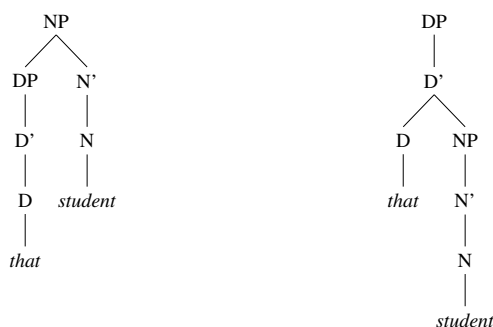
Although the determiners take no dependents of their own here, they nevertheless

vacuously project three levels of unary branching structure.<sup>8</sup> In MP, such structures are absent entirely because bar levels play no part in the theory. This is also true of the MG formalism presented here, although as noted the GB style Xbar trees are included in the MGbank corpus for convenience.

It is worth observing, however, that in mainstream Minimalist publications, BPS trees are standardly still annotated with category and bar level diacritics to make the trees easier to read, as illustrated for our example VP in fig 3.3; this tree is just a relabelling of the one in fig 3.1 and therefore lacks the vacuous unary branches of fig 3.2. We will often adopt this notation in the remainder of this dissertation for convenience.

Xbar Theory also stipulated that phrases could in principle contain at most one specifier and one complement. In early GB, specifiers included functional items such as determiners and auxiliaries, which cannot iterate, and so the stipulation of a single specifier seemed natural enough. In Minimalism, however, functional items such as determiners, auxiliaries and complementizers have been reanalysed as heads of the

<sup>8</sup> The motivation for vacuously projecting X' nodes was the observation that certain proforms (*one*, *do so* etc) appeared to substitute only for X' level constituents. For example, it is possible to say *that* [<sub>N'</sub> *student of physics*] *with the long hair* and *that* [<sub>N'</sub> *one*] *with the short hair*, but not *...and that one of chemistry with the short hair*, because *of physics/chemistry* is a complement (whereas *with the long/short hair* is an adjunct adjoined to a higher N'), and hence is contained, along with the head, in the lowest N' node; the head and the complement must therefore both be substituted for by *one*. Now consider the fact that it is perfectly acceptable to say *this student* and *that one*. This indicates that even when *student* does not take a complement, it still vacuously projects an N' node which can be targeted by *one* substitution. However, this argument fell apart once functional heads such as auxiliaries and determiners were reanalysed as heads rather than dependents. This meant that the structure on the left below was reanalysed as the structure on the right, in which *student* is now an NP meaning that *one* substitution can target NP rather than N'.



The so-called DP hypothesis was originally proposed in Abney (1987), and a good summary of the arguments in its favour is presented in Radford (2004). One theoretical motivation for adopting it was that determiners were now viewed as taking complements of their own. In this way, the functional head hypothesis, of which the DP Hypothesis is but one instance, allowed Xbar Theory to be fully generalised to all lexical items, functional as well as contentful.

In view of the fact that other syntactic operations such as movement never appeared to target X' nodes, these developments in the theory seriously undermined the status of bar levels as substantive primitives. See Hornstein et al. (2005) for a more in depth overview.

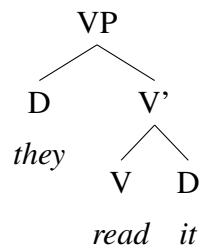


Figure 3.3: Derived Xbar Theoretic equivalent of the BPS tree in fig 3.1

nominal and verbal phrases containing them (see fn 8), and the term *specifier* is now used as a generalisation of the term *subject* (just as *complement* is a generalisation of the term *object*).<sup>9</sup> Examples of specifiers thus include subjects themselves, topicalised phrases, focused phrases and fronted wh-constituents etc, and it is far less clear that these types of constituents cannot iterate (for example, there exist multiple wh-fronting languages such as Bulgarian, Serbo-Croatian, Czech and Polish), thus undermining the single specifier claim.

As we have seen, in classical Xbar Theory, the different types of dependents were defined in configurational terms, via reference to the different bar levels. In Minimalism these definitions have become untenable, in part because bar levels themselves are no longer regarded as theoretical primes, but also because allowing multiple specifiers into the theory makes the definition of a specifier as daughter to the sole XP projection and sister to an X' unworkable (see Hornstein et al. (2005) for discussion of these points). In both MP and MG complements and specifiers are therefore, like bar levels themselves, derivative notions: a complement (*it* in fig 3.1) is simply the first dependent to be set Merged<sup>10</sup> with the head, whereas all subsequently set Merged constituents (*they* in fig 3.1) are specifiers. On this definition, a phrase can have at most

<sup>9</sup>Some Minimalists follow Kayne (1994) in conflating specifiers with adjuncts in the syntax. These authors therefore refer to modificational PPs, adverbs and adjectives as specifiers. The MGBank grammar does not follow this tradition, however.

<sup>10</sup>Chomsky (1995, 2001a) distinguishes standard set Merge, used for complements and specifiers, from pair Merge used for adjuncts. These two operations result in different types of constituent labels which are presumably used to distinguish adjuncts from other dependents for interpretive purposes. The motivation for this approach has partly to do with the fact that the GB definition of an adjunct as daughter and sister to X' (or daughter and sister to XP) is unworkable in Minimalism, owing to the fact that bar levels are not theoretical primitives. Additionally, the fact that not only adjuncts but also now specifiers can iterate makes these two types of dependent difficult to distinguish configurationally. As we shall see in section 3.3.1, in MGs adjunction also uses different rules and also different features from other instances of Merge. These features would be associated with different semantic values in the directional compositional approach to semantics of Kobo (2006), making adjunction straightforwardly distinguishable from standard Merge for interpretive purposes without recourse to phrase structure tree labels.

one complement (in accordance with classical Xbar Theory) but can in principle have any number of specifiers (contra classical Xbar Theory).

One obvious candidate for a construction with multiple overt specifiers in English is coordination (see, e.g., Zhang (2010) and section 3.3.7.1 below), if we assume that the coordinator is the head of the coordinate phrase with the conjuncts its ‘arguments’. Consider the fact that only a single post-coordinator conjunct is allowed in English, whereas an unlimited number of pre-coordinator conjuncts is possible. This suggests an analysis in which the post-coordinator conjunct is a complement and all pre-coordinator conjuncts are specifiers. This multiple specifier analysis is illustrated in 3.4 for the phrase *Tom, Dick and Harry*.

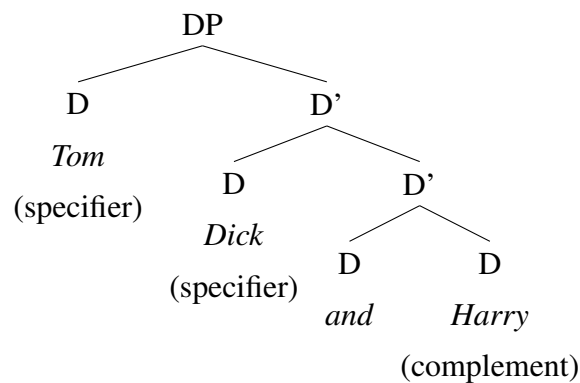


Figure 3.4: BPS tree with multiple specifiers for the coordinate phrase *Tom, Dick and Harry*. In Zhang (2010), the conjunct inherits its category (here D) from the leftmost conjunct. In the MG presented here, this inheritance is simply precompiled into the lexicon.

Finally, consider the Xbar Theoretic stipulation that all dependents are maximal XPs. In MP this is regarded as an inevitable consequence of the fact that in order to establish a local syntactic relation such as spec-head or head-comp, the dependent must be immediately contained within a projection of the head. Given Minimalism’s relational definition of a maximal XP projection as a syntactic object that does not project, the maximal status of all dependents immediately follows. This is clearly a representationally-oriented perspective which can also be adopted in an MG context. However, in MGs this is also another consequence of the strict ordering of syntactic feature types: to say that a dependent is non-maximal would be to say that it has argument positions left to fill, and in an MG framework that it has selectors and/or licenser features left to check and delete. However, given that a constituent can only become a dependent once its selectee feature is exposed, and given that the selectee feature necessarily follows any selectors and licensors, the maximal status of all dependents is

guaranteed.

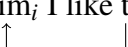
### 3.2.3 Phrasal Movement

Our earlier derivation involved establishing only local (spec-head and head-comp) dependencies in which each dependent was immediately contained within some projection of the head. However, there are many constructions in natural language which involve dependencies over greater structural distances. The defining property of Minimalism setting it apart from other grammatical formalisms, such as CCG, TAG, LFG and HPSG, is its use of transformational movement operations to capture many of these dependencies. Consider the following dialogue.

- SPEAKER 1: Do you like Jack?  
 SPEAKER 2: No, not really.. he's pretty bossy most of the time.  
 SPEAKER 1: What about Pete?  
 SPEAKER 2: *him*, I like.

The final line of this dialogue features an instance of *topicalisation*, in which the object *him* does not appear in its canonical post-verbal object position, but instead appears in the left periphery of the clause, where it is offset by a special intonation (indicated by the italics above). Topicalisation is generally used to emphasise information that was already introduced earlier in the discourse, and in MP it is treated as an instance of movement of the topicalised element from its canonical base generated position to the left periphery of the clause. Although in this case the movement is of a single word, it is perfectly acceptable to move an entire phrase, as in *that man, I like*. Topicalisation is therefore regarded as an instance of phrasal movement.

In GB, the transformational rule  $\text{move-}\alpha$  mapped the Deep Structures created by Xbar Theoretic PS rules onto Surface Structures by moving constituents from their base positions into structural slots that had already been generated at DS.  $\text{Move-}\alpha$  left behind silent traces in the original positions of the mover, and these were co-indexed with their overt antecedents, as illustrated for our example sentence below.

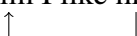
- (9)  $\text{him}_i$  I like  $t_i$
- 

Traces and indices were not merely notational conveniences intended to enhance the readability of trees, but were treated instead as substantive syntactic objects. For example, in Chomsky (1981) traces have features and requirements which are distinct from those of their antecedents; there are also different types of trace, each with their

own distinct features. For example, all traces must be *properly governed*, but while A-movement traces are +anaphor, A'-movement traces are -anaphor. However, Chomsky (1995) argued that traces and indices should be eliminated from the theory entirely owing to the fact that these items, like bar levels, violate the Inclusiveness Principle. On the other hand, moved items often appear to be interpreted in some lower position which they have occupied during the course of the derivation, and for Chomsky, this fact argues against an analysis in which moved items simply leave behind an empty node.

To resolve this issue, Chomsky (1995) argues for a *copy theory of movement* according to which the operation Move is treated as a composite operation consisting of the two sub-operations Copy and Merge. When a constituent moves, the subtree in which it is rooted is first copied, and then the copy is merged with the root of the tree (hence Move can be regarded as Internal Merge). A separate algorithm at PF then determines which of the two copies is to be phonetically realised (see Nunes (2004) for detailed proposals regarding this aspect of the theory), though both are potentially interpretable at LF. Phonetically deleted copies are generally indicated with strikethrough text, as illustrated below.

(10) him I like ~~him~~



In MG derived trees, the residue of movement is usually indicated with  $\lambda$  symbols which ostensibly appear more like traces than full copies. However, these items lack the indices of GB traces and are absent entirely from the derivational structure. As we shall see in the next section, within the MG community a consensus has emerged over the past ten years or so which views derived phrase structure trees as artefacts of the derivation, with the derivation tree now being regarded as the primary syntactic data structure. From this perspective, the question of the status of the  $\lambda$  traces in MG derived trees is arguably not a coherent one, because these items do not exist in the derivation tree, hence they do not exist in the syntax at all (though they do, of course, exist as variables in the semantic logical form). We shall return to the status of the copy theory of movement in MGs in section 3.2.5 after introducing derivation trees.

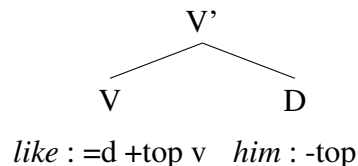
### 3.2.4 A derivation involving Move

In order to derive our topicalisation sentence, we will start with the following lexicon.

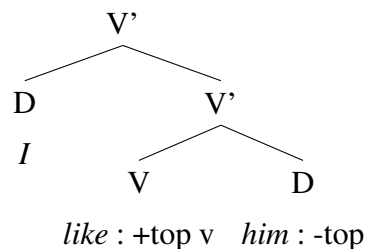
like :: d= =d +top v

I :: d  
 him :: d -top

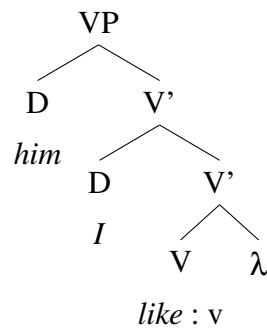
The derivation proceeds as follows. First, we check and delete the  $d=$  feature of *like* and the  $d$  feature of *him*, by merging *him* to the right of *like*. This results in the following  $V'$  constituent:



The head of this phrase is *like* by virtue of the fact that it has an active  $=d$  feature indicating that this expression must next be Merged with a DP on its left. Notice that, unlike in our earlier derivation, the object still has a remaining licensee feature and hence is still active in the derivation. We next merge *I* to the left of *like him*, checking both the  $=d$  feature of *like* and the  $d$  feature of *I*. This results in the following constituent which is also informally definable as a  $V'$  owing to the fact that the head still has an undeleted licensor feature.



Licensor features cannot standardly be checked via External Merge in the MG formalism, but must instead be checked via Move (Internal Merge). This means that the head of some expression whose active feature is a licensor must have that feature checked by one of its moving subparts with a matching licensee feature. Chomsky (2000) refers to the head with the licensor feature as a *probe* and the constituent with the matching licensee as the *goal*. In our case, the probe is *like* and its goal is *him*. Feature checking in both Chomsky (1995) and MGs can only be performed when the probe and the constituent headed by the goal enter into a local configuration, such as spec-head or head-comp. In the case of feature checking via movement, spec-head is the only available option because the probe necessarily already has a complement. We will therefore check and delete the  $+top/-top$  features by moving *him* to become a second specifier of *like*, thus forming the VP shown below.



Notice that the fact that *like* is a verb is represented twice in the above structure, once by the *v* feature and once by the V label for this terminal. This illustrates why in Minimalism even category labels are considered redundant. This redundancy is eliminated as soon as we convert to the BPS-like MG derived tree notation, as shown in fig 3.5.

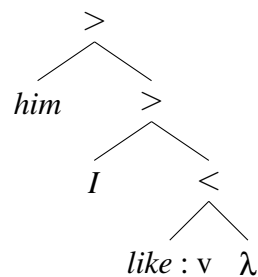


Figure 3.5: A simplified MG derived tree for the sentence *him, I like* which features topicalization of *him*.)

Notice that in this structure the probe c-commands<sup>11</sup> the  $\lambda$  trace, as is standardly assumed to be the case for all (non-remnant, non-sideward) movement in MP. C-command has sometimes been regarded in TG as the fundamental long-distance syntactic relation, or at least configuration, since it is implicated not only in movement operations, but also in other long-distance relations such as binding, control and polarity item licensing, etc.

Hornstein (2009) argues that c-command is best regarded as a derivative concept (a configuration rather than a relation) which falls out, in the case of movement, from the fact that derivations must obey Chomsky's (1995) Extension Condition (EC). This condition states that syntactic operations are strictly monotonic in that they must always extend the root of the tree. The Merge and Move rules presented in this dissertation all adhere to this condition (i.e. there is no 'tucking in' in the sense of Richards

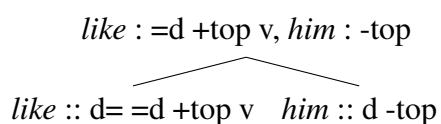
<sup>11</sup>The standard definition of c-command is that a head c-commands its sister node and all of its sister's descendants. Hence in our example *I* commands the inner *V'*, the *V* and the  $\lambda$ , for instance.



(1997)). The c-command requirement between antecedents and traces therefore follows here from EC, as it does for Hornstein. Furthermore, MGbank also follows Hornstein (2001) in treating control and binding as instances of A-movement; polarity item licensing of words like *any* and *much*, meanwhile, is treated as covert movement. The c-command ‘requirement’ for these other types of relation therefore also holds in MGbank and again derives from EC.

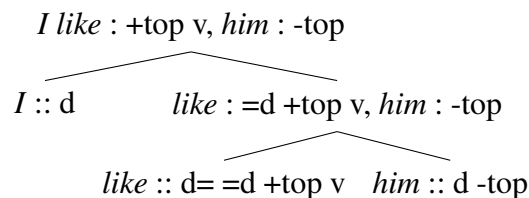
### 3.2.5 MG derivation trees

Over the past decade and a half, a shift in perspective has occurred within the MG research community so that derived phrase structure trees are no longer viewed as the primary syntactic data structure. In fact, such structures seem to be viewed by most MG researchers as being merely artefactual. Instead, the primary syntactic object of interest is now considered to be the MG derivation tree, which records the structure of the derivation itself. For example, consider again our example sentence *him, I like* from the previous section. The generation of a derivation tree for this sentence proceeds as follows. The first operation is the merger of *like* and *him*, checking the *d=* feature of the former and the *d* feature of the latter. Because this is an instance of External Merge, it is necessary to specify two arguments to this operation, the selecting and the selected expression. We must therefore represent this Merge operation with a binary branching derivational structure, with the two lexical arguments at the leaves and the result of the Merge operation at the root.



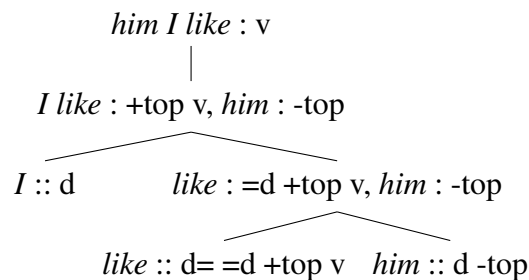
Notice the comma separating the two subcomponents, or *chains*, of the expression at the root of this tree. The first chain in any given expression is the head of that expression, whereas all other chains are movers (their ordering being irrelevant).

The next operation is another instance of External Merge, as there are again two arguments. The selecting expression is that which was derived in the previous Merge step, while the selected item is the pronoun *I*. The Merger of these two items can again be represented as a binary branching node, shown below, with the result at the root.



Notice that because *I* had no more features to check after its *d* selectee was deleted, its head string was concatenated with the head string of the selecting constituent.

So far the derivation tree we are generating has precisely the same geometry as its equivalent derived tree. However, the next step is to move *him* to the specifier position of the main structure, an instance of Internal Merge. Owing to Stabler's strict version of the Shortest Move Constraint (introduced in the previous chapter), there can only ever be one moving chain with a given licensee feature as its first feature. It is therefore unnecessary to specify the licensing and licensed chains as arguments to Merge; instead, we can simply specify the entire expression as the sole argument to Move. Although Move therefore results in binary branching phrase structures as we have seen, at the level of *derivational* structure it can be represented by a unary branching node as follows.



The above tree now differs crucially in geometric terms from the earlier derived tree owing to the fact that here all the terminal nodes remain in their base generated positions. This is a consequence of the fact that movement is not represented in the derivation tree by the displacement of nodes, but is instead represented solely by the reordering of the string components of expressions at each non-terminal relative to those non-terminals which it dominates.

Notice that as well as the reordering of strings being represented at the non-terminals of the derivation tree, so too were the various stages of feature checking operations. In fact, all information need to continue driving the derivation at any given stage is encoded in the root of the partially built tree. In effect, each non-terminal can be viewed as a compact representation of a phrase structure tree, or as a *collapsed tree* in the terminology of Stabler (2001a). Each collapsed tree contains all and only the syntactic

information needed to continue driving the derivation, with any unnecessary geometric details being disregarded. For example, the two partially constructed derived trees in figs 3.6 and 3.7 are identical from a purely syntactic and derivational perspective (the case features have been added here to ensure that the correct word order could ultimately be achieved for 3.7. See section 4.7.2 on Case Theory).

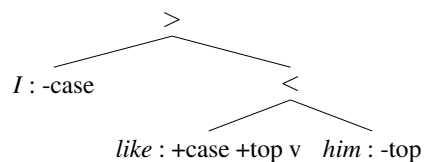


Figure 3.6: A partially constructed MG derived tree for the sentence *him, I like*.

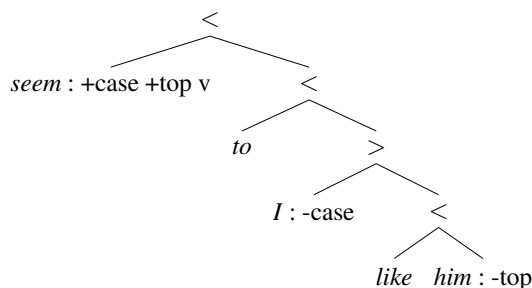


Figure 3.7: A partially constructed MG derived tree for the sentence *him, I seem to like*.

The tree on the right clearly contains a good deal more internal phrase structure than that on the left. However, because both trees will exhibit identical derivational behaviour from here on out, in the MG formalism they share the following abstract collapsed tree representation (or type), where *s*, *t* and *r* are abstract placeholders for the phonetics of the three chains.

$$s : +\text{case } +\text{top } v, t : -\text{top}, r : -\text{case}$$

Mainstream MP practitioners may object at this point that the internal geometry of the phrase structure tree is required for interpretive purposes. After all, it clearly matters at the level of semantics whether or not the topic of the sentence was base generated as a subject or an object. Furthermore, recall that in early MP there were considered to be two levels of linguistic representation, LF and PF. LF was a phrase structural representation which included all semantically relevant information (including covert movements) and was interpreted by the C-I performance systems. In the later probe-goal/Agree-based Minimalism, Chomsky argues that syntactic derivations proceed incrementally by constructing *phases* (vP and CP), whose complements are transferred directly to the interfaces as soon as they are completed. (Chomsky, 2004, page 152) argues that this cyclic property allows us to dispense with LF as a linguistic level of representation, leaving us with just the semantic interface and PF (which here

he appears to regard as synonymous with the A-P interface itself, though how this can be so given that A-P is a performance system is unclear). However, it seems from the following quotation that Chomsky still regards phases as phrase structural objects which must undergo interpretation.

*Of all those levels, the only ones that remain are PF and the semantic interface. The others, the strictly internal linguistic ones - LF, d-structure, s-structure, probably don't exist...At some point in the derivation, you've got a syntactic object, call it a 'phase' by definition. That syntactic object is handed over to the interpretive systems; it's transferred to the phonology and the semantics. They do whatever they do to it, and then they're finished with it.'*

(Chomsky, 2011, page 152)

In the early days of MG, the derived phrase structure tree was also regarded as the primary syntactic object and the input to interpretation. A more recent trend within this framework, however, has been to view MGs as generators of derivation trees, rather than phrase structure trees, with the derivation tree undergoing interpretation directly. This point is made explicit in the following quotation from Graf et al. (2017).

*Since derivation trees provide a record of how a given phrase structure tree is to be assembled, they implicitly contain all the information encoded in the latter. In itself this is a rather unremarkable fact, but in the MG community a trend has developed in the last 10 years to treat derivation trees as the primary data structure of MGs. That is to say, MGs are no longer viewed as generators of phrase structure trees or strings but rather as a generator of derivation trees.*

(Graf et al., 2017, page 8)

However, Kobele (2006) adopts a subtly different perspective, showing that it is possible and indeed methodologically preferable to view the derivation tree itself simply as a trace of the process which derives the pure PF and LF objects in a strongly cyclic fashion. On this view, MGs are generators of strings and logical forms, rather than of either derivation or phrase structure trees. This is essentially the perspective on syntactic structure argued for in Steedman (1996).

*...syntactic structure is merely the characterization of the process of constructing a [form-meaning pair], rather than a representational level of structure that actually needs to be built...*

(Steedman, 1996, page xi)

Kobele (2006, page 55) is explicit about his aim to ‘see how this project can be realised in minimalist grammars.’ Consider the PF representation first. Notice that in the derivation presented above, the root node already encodes the precise linear order of all the terminals, and that this PF representation was constructed in strictly cyclic fashion in lockstep with the syntactic derivation. There is thus no need to construct a phrase structure tree in order to determine linear order.

On the LF side, (Kobele, 2006, page 62) presents what he refers to as a ‘direct compositional’ approach to semantics for MGs in which each syntactic feature is associated with a semantic value, which may be the identity function for semantically vacuous syntactic features. These semantic values then interact with one another as each syntactic checking operation takes place, for example via semantic operations such as functional application. Under this approach, therefore, the root node of any MG derivation tree already contains the PF and LF representations for the portion of the derivation that has been processed up to that point. There is thus no need for any level of syntactic *representation* as such: syntax is viewed here instead simply as the *procedure* that was used to derive the pure phonetic representation (i.e. the string), and the pure semantic representation (e.g. a first order logical representation). In terms of its relation to the proposals made in MP, this architecture is most similar to the multiple Spell-Out model of Chomsky’s Phase Theory, seen in fig 2.3 in the previous chapter. However, unlike in Chomsky’s system, interpretation does not wait until each phase head is reached, nor does it involve transferring small chunks of phrase structure to the interfaces for interpretation/linearisation. Instead, interpretation and linearisation both proceed in lockstep with each syntactic operation, with phrase structure, and even derivational structure, essentially viewed as (2nd and 1st order) traces of this process.

On this perspective, the fact that the collapsed tree representation eliminates much of the geometry of the phrase structure tree is inconsequential because the semantic (and linear) information which it is intended to encode is in fact much more directly and immediately encoded in the collapsed tree non-terminals of the derivation tree.

The switch in perspective to focusing on derivation trees as opposed to phrase structure trees also has important consequences for MG parsing. For example, the fact that the nodes of the derivation tree do not undergo movement means that the string languages at the fringes of any well-formed MG derivation trees are context free (and the set of MG derivation trees is therefore regular). This property has enabled a number of well-studied polynomial time context free parsing algorithms, such as the CKY and Earley variants in Harkema (2001), to be adapted to MGs, and in fact all

MG parsers of which I am aware, including the ones constructed for this dissertation, generate derivation trees, rather than phrase structure trees directly. However, because all of the information needed to build the phrase structure tree is implicitly encoded in the derivation tree, the former can be derived deterministically via a single traversal of the latter using a multiple bottom-up tree transducer (Stabler, 2013a, page 6). Hence the name *MG derived tree* to describe MG bare phrase structure trees.

It is also worth noting that several contentious issues within MP simply evaporate once derivation trees are viewed as primary. Take the question of labelling, for example. Chomsky (2013) regards the *labelling* of non-terminals in the phrase structure with the head item as a distinct operation in the grammar which in some cases actually forces (successive cyclic movement). In Chomsky (1995) different types of label were used to distinguish the adjunction from other types of Merge, owing to the fact that it was no longer possible to distinguish specifiers from adjuncts via reference to bar levels (and both could now iterate freely). On the other hand, Collins (2002) has argued for a label-free syntax, and it has been suggested that (depending on how the operation is formulated) labels may be in violation of Chomsky's own Inclusiveness condition as they introduce items into the derivation which were not in the lexicon (Hornstein et al., 2005, page 207). Once phrase structure trees are eliminated from the grammar, however, the question of the labelling of phrase structure tree non-terminals simply does not apply.

Another question which has been much discussed in the GB and MP literature concerns the status of the traces left by movement. In GB these items were considered to be substantive theoretical primitives with their own unique sets of features differing from those of their antecedents. Chomsky (1995) argued that such items should be banned because they violate his Inclusiveness Condition which bans grammatical formatives which are not contained in the lexicon. Instead, he proposed his Copy Theory of movement, according to which traces are in fact copies of their antecedents which are deleted at PF. This idea has been widely adopted within Minimalism and is perhaps most fully worked out in Nunes (2001), but it has also been criticised, e.g. by Cormack and Smith (2002) and Epstein and Seely (2006).

Notice however, that if nodes do not move at all, as is the case in MG derivation trees, then there can be no question about what they leave in their wake. The purpose of traces is clearly to encode that one constituent can be interpreted in multiple syntactic positions. As such they are essentially book-keeping devices which record certain semantically relevant derivational steps. Traces and the phrase structure trees which

contain them are certainly convenient representations for linguists to work with, but this convenience has nothing to do with their psychological reality. Moreover treating them as real can actually be harmful (if the strong derivational approach adopted here is correct), since it leads to theoretical proposals being made which rely on various purported properties of these constructs. Regarding derivation trees as the primary syntactic data structure forces us to eliminate traces/copies altogether, and is arguably more ‘Minimalist’ than the Copy Theory as it results in less theoretical postulates. Instead, traces are restricted precisely to appearing only where they truly are ‘virtually conceptually necessary’, which is as variables in the semantic logical form. This is a popular perspective within the MG research community, but is much more marginal in mainstream MP. One notable exception is Epstein and Seely (2006), who argue for the primacy of the derivation itself in the following two quotations.

*Chain theory and trace theory are annotational look back devices encoding derivational history...whereby the (empirically important) derivation that produced the output representation is encoded.*

(Epstein and Seely, 2006, page 45)

*..trace theory is a representational ‘coding trick’ necessitated by the failure to recognise the importance of the constrained application of explicitly defined rules themselves (the derivation)...we would assume that there are no traces, not even the chain-tail trace. The information represented by the trace tail of the chain is already part of the derivation, e.g. for direct objects the relation is expressed by ‘Concatenate/Merge theta marker V and DP’.*

(Epstein and Seely, 2006, page 46)

Because the MG derivation tree records all of the merge and move operations which have applied, it encodes all of the information which traces are intended to encode without the need for such ‘representational coding tricks’. The pure semantic representation can therefore either be read off of the derivation tree, or constructed in tandem with it as in Kobele (2006)<sup>12</sup> and categorial grammar approaches. This latter option

<sup>12</sup>Interestingly, Kobele (2006) argues for the existence of phonetic copying operations in the West African language of Yoruba. Phonetic copying can straightforwardly be captured in an MG by allowing a string variable occurring on one of the arguments to a Merge/Move rule to appear more than once in the output of the rule. Allowing for this operation increases the weak generative power of the formalism to that of Parallel Multiple Context Free Grammars (PMCFGs), which are a superclass of MCFGs that are not mildly context sensitive but still polynomial. Whether or not natural languages actually require

seems to me to be preferable, since it allows us to dispense entirely with the idea that there are any syntactic representational objects whatsoever.

It may be objected that derivation trees have labels, and therefore that the question of how labels are created still needs an answer. This would be to miss the crucial point, however, which is that even the derivation tree need not be viewed as a substantive psycholinguistic object. Instead, it can simply be regarded as a convenient depiction of the history of the operations which must be applied by a parser in order to build the derived phonetic and semantic representations. In fact, the non-terminals of derivation trees in the MG literature are often not annotated with collapsed trees at all, but rather with labels which encode the particular operations which applied in order to form them. For example, our earlier derivation tree example is sometimes represented as in fig 3.8, where black dots indicate Merge and white dots indicate Move, and sometimes as in fig 3.9 where these operation names are included explicitly. Provided there is enough information in these labels to deterministically recover the intended derivation, the actual labels used are unimportant.

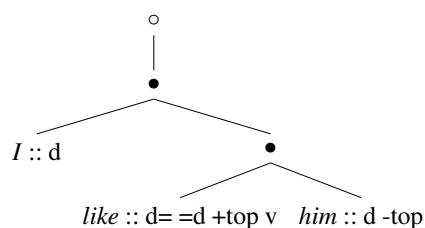


Figure 3.8: An MG derivation tree with dot notation often used elsewhere in the MG literature. Black dots indicate a Merge operation, white dots indicate a Move operation.

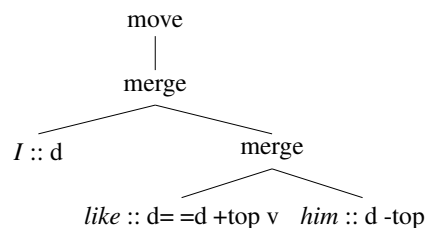


Figure 3.9: The same MG derivation tree represented with explicit operation names at non-terminals. This type of derivation tree is included in MGbank.

Because the MG presented in this dissertation includes considerably more rules than in the simple MGs usually discussed in the literature, the format in fig 3.9 is included alongside the derivation trees with collapsed tree non-terminals in MGbank.

### 3.2.6 Formal definition of a Directional Minimalist Grammar

A Directional Minimalist Grammar is defined as a quadruple  $(\Sigma, Cat, Lex, F)$  s.t.:<sup>13</sup>

these phonetic reduplication operations has been contested in Pullum and Rawlins (2007), however.

<sup>13</sup>In order to unify the notation for merge and move, we adopt the convention that all diacritics appear on the side of the Part of Speech (PoS) symbol on which selection occurs; hence  $x=$  indicates rightward



1.  $\Sigma = P \cup I$  is a finite set of non-syntactic features ( $P$  = phonetic features,  $I$  = semantic features).
2.  $Cat = selectees \cup selectors \cup licensees \cup licensors$  is a finite set of syntactic features, s.t. for each feature  $x \in selectees$  there are features  $(=x, x=) \in selectors$ , and for each feature  $-y \in licensees$  there is a feature  $+y \in licensors$ .
3.  $Lex$  is a finite set of axioms (lexical items) over  $V \cup Cat$ , with the  $Cat$  features on each simplex tree strictly ordered from left to right.
4.  $F$  is a set consisting of the structure building functions  $MERGE$  and  $MOVE$  (the deductive rules of inference), defined as the union of their respective sub-functions, given in figures 3.10 and 3.11, where expressions are contained within square brackets, chains are separated by commas,  $\alpha_1, \dots, \alpha_k$  is a (possibly empty) set of moving chains,  $\delta$  and  $\gamma$  are feature sequence suffix variables, with  $|\delta| \geq 1$  and  $|\gamma| \geq 0$ ,  $s$  and  $t$  are string variables, and string/feature separators indicate whether a chain represents an unmerged lexical head ( $::$ ) or a derived element ( $:$ ), or can be either ( $;$ ).

$$\frac{[s::x=\gamma] \quad [t;x, \alpha_1, \dots, \alpha_k]}{[st:\gamma, \alpha_1, \dots, \alpha_k]} (merge1(comp))$$

$$\frac{[t;x, \alpha_1, \dots, \alpha_k] \quad [s::=x\gamma]}{[ts:\gamma, \alpha_1, \dots, \alpha_k]} (merge2(comp))$$

$$\frac{[t;x] \quad [s:=x\gamma, \alpha_1, \dots, \alpha_k]}{[ts:\gamma, \alpha_1, \dots, \alpha_k]} (merge3(spec))$$

$$\frac{[s::x=\gamma] \quad [t;x\delta, \alpha_1, \dots, \alpha_k]}{[s:\gamma, t:\delta, \alpha_1, \dots, \alpha_k]} (merge4(comp))$$

$$\frac{[t;x\delta, \alpha_1, \dots, \alpha_k] \quad [s::=x\gamma]}{[s:\gamma, t:\delta, \alpha_1, \dots, \alpha_k]} (merge5(comp))$$

$$\frac{[t;x\delta] \quad [s:=x\gamma, \alpha_1, \dots, \alpha_k]}{[s:\gamma, t:\delta, \alpha_1, \dots, \alpha_k]} (merge6(spec))$$

Figure 3.10: Sub-functions of MERGE

---

selection, =x leftward selection.

$$\frac{[s: +f \gamma, \alpha_1, \dots, \alpha_{i-1}, t: -f, \alpha_{i+1}, \dots, \alpha_k]}{[t s: \gamma, \alpha_1, \dots, \alpha_{i-1}, \alpha_{i+1}, \dots, \alpha_k]} (\text{move1})$$

$$\frac{[s: +f \gamma, \alpha_1, \dots, \alpha_{i-1}, t: -f \delta, \alpha_{i+1}, \dots, \alpha_k]}{[s: \gamma, \alpha_1, \dots, \alpha_{i-1}, t: \delta, \alpha_{i+1}, \dots, \alpha_k]} (\text{move2})$$

Figure 3.11: Sub-functions of MOVE

For a given Minimalist Grammar  $G = Lex$ , the language  $L(G)$  is the closure of  $Lex$  under the structure building functions  $\{MERGE, MOVE\}$  in accordance with the Shortest Move Constraint (to be discussed in section 3.2.10):

**The Shortest Move Constraint (SMC):** Two licensee features may both be active at the same time in the derivation only if they are distinct.

In fig 3.10, Merge is split into separate subrules according to whether the dependent is selected to the right (Merge1/Merge4) or left (Merge2/Merge3/Merge5/Merge6) of the governor, and also according to whether the selectee has additional licensee features to check and hence must start moving (Merge1/Merge2/Merge4/Merge5) or whether it has no more licensee features, in which case its string will simply be fused onto that of the governor (Merge3/Merge6). Notice too that in these rules we are assuming that while complements may contain movers (Merge1/Merge2/Merge4/Merge5), specifiers may not (Merge3/Merge6) (although the specifier itself may move of course (Merge6)). As stated here, these rules therefore enforce a strict version of the Specifier Island Constraint (SpIC) (Huang, 1982). However, as we shall see in section 4.3.1.1, it did not prove possible to maintain the strong version of SpIC for the MGBank grammar, given its doubling analysis of constructions involving overt anaphors or associates (such as reflexive bind and floating quantifier constructions).

As fig 3.11 shows, Move is also divided into subrules according to whether the moving item must move again (Move2) or whether it has no more licensees and hence does not move any further (Move1).

### 3.2.7 Null Heads and Basic Clause Structure

Until this point, all of the lexical categories we have looked at have included all three basic types of features: syntactic, phonetic and semantic (though we are leaving the

semantic features implicit here). However, it is a logical possibility that there could be lexical items which lack syntactic, phonetic, and/or semantic features entirely.

An example of lexical items which could be argued to have semantic and phonetic features but (at least in certain contexts) lack syntactic ones are interjections, such as *wow!*, *no!*, *ouch!* etc, since these items do not seem to participate in syntactic derivations, at least outside of quotative contexts (“*no!*” *he said.*). There also appear to be lexical items which have syntactic and phonetic features, but lack semantic ones: pure case-marking prepositions, such as *of* in *student of physics*, are one obvious candidate, on the standard assumption that structural case is semantically vacuous.

Minimalists also argue for the existence of lexical items which contain syntactic and semantic features, but lack phonetic ones. In this section we will look briefly at some of the motivation for assuming phonetically null heads to exist (for more extensive arguments, see Radford (2004, chapter 4)). We will then move on to looking at the basic universal clause structure which is assumed in Minimalism and which presupposes the existence of phonetically null heads. A more detailed examination of each of the three clausal domains (VP, TP and CP) will be undertaken in section 4.7 after the introduction of the various extensions to the MG formalism (such as head movement) in section 3.3.

To begin the discussion of null heads, consider the following paradigm (taken from (Radford, 2004, page 124)):

- (11) a. We didn’t know [if he had resigned].  
 b. We didn’t know [that he had resigned].  
 c. We didn’t know [he had resigned].

The embedded clause in 11a clearly has interrogative force, while that in example 11b has declarative force. The difference in meaning is clearly related to the different choice of complementizer (*if* vs *that*) in the two cases. In the direct compositional approach to semantics of Kobele (2006) which we are assuming here, this can be interpreted as meaning that there are two different semantic values attached to the *c* selectee of these complementizers. Notice, however, that example 11c can only be declarative, not interrogative, and yet there is no overt complementizer here meaning that there is ostensibly no *c* selectee feature to which we can attach whichever semantic feature encodes declarative force. One approach would be to simply assume that declarative is a default value that is assigned to a clause unless another illocutionary force is indicated. An alternative perspective, which is widely adopted within the mainstream Minimalist

community and will be pursued here, is to assume that in fact there is a null declarative complementizer (a silent counterpart of *that*) in 11c which encodes declarative force. Thus 11c has the following basic structure.

- (12) We didn't know [<sub>CP</sub> [<sub>C</sub> [decl]] he had resigned].

Here, [decl] indicates an empty string corresponding to the null *c* head. The empty strings of phonetically null lexical items are annotated using this square bracket notation in MGBank in order to make the trees easier to read. In all other instances (such as at the non-terminals of the derivation trees), however, the empty string is represented using  $\epsilon$ .

The CP layer projected by the embedded null C head (or, from an MG derivational perspective, its *c* selectee feature) enables the clause it heads to be straightforwardly selected by verbs requiring a CP complement, such as *know* or *say*. Note also that because *main* clauses also bear declarative, interrogative, imperative or exclamative force, Minimalists assume that they too contain a complementizer in their left periphery, though in this case the complementizer is obligatorily null.<sup>14</sup>

Notice that in 12 the complementizer is assumed to head the embedded clause. One of the most significant developments within GB theory during the 1980s was the development of the hypothesis that functional elements such as auxiliaries, complementizers and determiners headed their own phrases which contain the lexical items of which they were previously thought to be (specifier) dependents. This allowed Xbar theory to be successfully generalised to all lexical items, both functional and lexical (Chomsky, 1986a). Thus the determiner heads the DP nominal phrase, taking the NP headed by the noun as its complement (Abney, 1987). The complementizer, meanwhile, as the highest functional head in the clause, is standardly viewed as the syntactic head of that clause (Stowell, 1981; Chomsky, 1986a).

This perspective is also adopted here. The reader will recall that in order for an MG derivation of a sentence to converge it must be the case that the only remaining feature in the tree is a single *c* (i.e. a complementizer selectee) feature on the head of that tree. This is equivalent to saying that a main clause must project a CP layer that does not have any licensee features to check via movement. CP is thus equivalent to the S node (or Sbar when present) in a traditional phrase structure grammar. All the derivations we have looked at so far were thus incomplete because they only projected

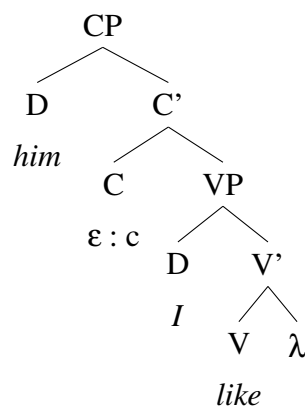
<sup>14</sup>For a review of the many and varied empirical arguments for the existence of phonetically silent lexical items, see chapter 4 of Radford (2004).

as far as VP. To remedy this, we will modify our analysis for the sentence *him, I like* by adding an entry to our lexicon for a null complementizer. We will also remove the +top feature from the verb and place it instead on this null C head in order to ensure that a topicalized phrase will move to the true left periphery of the clause.

like :: d= =d v

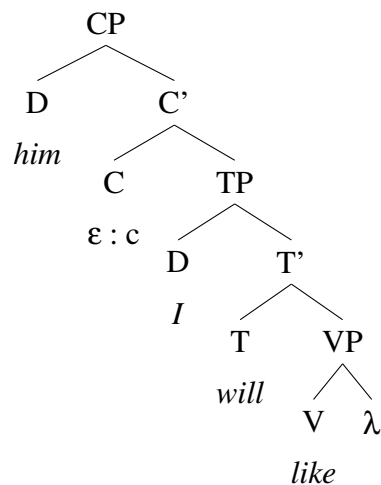
$\epsilon$  :: v= +top c

Using these categories, we can now derive the following CP for our example sentence:



This structure satisfies the formal requirement for a complete sentence that it be rooted by a CP with no remaining licensee features. However, Minimalists generally assume that clauses contain considerably more internal structure than this. For example, at present the above tree contains no head slot for an auxiliary verb, meaning that we are unable to derive a sentence such as *him, I will like*. In order to accommodate modals and other tensed auxiliaries, it is standardly assumed that between the CP and VP projections there is a TP projection whose T head is the locus of the tense properties of the clause and is either null<sup>15</sup> (in finite clauses lacking an auxiliary) or is host to a finite auxiliary (or, for infinitival clauses, the infinitival particle *to*). However, one problem is that so far we have assumed that subjects are generated as specifiers of the main verb, and yet clearly they appear on the surface to the left of any auxiliaries. One option, would be to suppose instead that they are actually specifiers to the tense phrase, and hence that our topicalization example has the following structure.

<sup>15</sup>In a full morphosyntactic theory, T initially hosts the tense suffix (e.g. *-ed* for past tense) which in the absence of an auxiliary in T must undergo a type of lowering head movement known as *affix hopping* which suffixes it onto the end of the main verb. Stabler (2001b) shows how a version of affix hopping can be incorporated into MGs, although this process is not currently modelled in MGbank.



This was in fact the standard analysis of the position of subjects in early GB. There is certainly good evidence that in English (and perhaps all languages) the verb and its object form a constituent which excludes the subject (and any auxiliaries). For example, whereas the object receives its thematic role directly from the verb, the thematic role of the subject appears to be determined in part by the object, where one is present, as the following examples illustrate.

- (13) a. *He took some money.*  
 b. *He took a rest.*  
 c. *He took the train.*

Furthermore, the verb and its object can be substituted for a proform like *do so*, leaving behind the subject (14a), whereas there are no proforms which target the subject and the verb but leave behind the object (14b).

- (14) *Jack ate apples in the morning...*  
 a. *...and Pete did so in the afternoon.*  
 b. *\*...and did so pears in the afternoon*

It is also interesting to note that the verb and its object can together become fossilised into an idiom which excludes the subject, while conversely, there do not appear to be idioms which are formed from the subject and the verb but exclude the object. Thus one can say *Pete/She/Tolkien kicked the bucket*, but not *The bucket kicked Pete/her/Tolkien*. These sorts of observations led researchers to assume that AGENT subjects were base generated outside of the verb phrase. For this reason, they became known as *external arguments*, in contrast to the *internal arguments*, such as the object, which are generated directly within the VP.

Minimalists still assume that the surface subject sits in spec-TP. However, throughout the 1980s and early 1990s, a number of authors (e.g. Zagona (1982), Kitagawa (1986), Speas (1986), Contreras (1987) and, most notably, Koopman and Sportiche (1991)) argued persuasively that although AGENT subjects do indeed appear in spec-TP in the surface structure, they actually start out their derivational lives within the verb phrase, and only later move to spec-TP to be assigned case (in GB) or check case (in early Minimalism).<sup>16</sup> This idea is often referred to as the Verb Phrase Internal Subject Hypothesis (VPISH), and is adopted by the MGBank grammar. We will therefore update our lexicon as follows.

$\epsilon :: t= +top\ c$   
 will :: v= +case t  
 like :: d= =d v  
 I :: d -case  
 him :: d -top

These lexical items will allow us to derive the structure in fig 3.12 for the sentence *him, I will like*, in which both the subject and the object originate inside the VP and subsequently undergo movement to check case and topic features respectively (in a more realistic grammar, such as MGBank, the object would also have a -case licensee which must be checked - see section 4.7.2 on Case Theory).

For good reviews of the many empirical arguments in support of VPISH (and the more general Predicate Internal Subject Hypothesis), which has become the standard assumption within MP, the reader is referred to Hornstein et al. (2005, pages 81-90) and Radford (2004, pages 241-250). In the remainder of this section we will briefly look at three of these arguments, two of which are empirical, with the other being a theory-internal argument from semantics.

The first empirical argument for VPISH comes again from idioms. Although an idiom formed from a subject and a verb which excludes the object is apparently impossible, there are many cases of idioms which are formed from the combination of the subject, verb and object, one example of which is the expression *the shit hit the*

---

<sup>16</sup>Note that although in English case only shows up morphologically in the personal pronoun system (e.g. nominative *he* vs accusative *him* vs genitive *his*), Minimalists assume that all DPs bear abstract structural case (at least those which function as arguments). In the TG literature, one often finds structural Case being written with a capital 'C' to distinguish it from nonstructural (i.e. inherent or lexical) case with a lowercase 'c'. For example, genitive case in English is sometimes treated as an inherent (i.e. theta-related) case. We do not make this distinction here and so will continue to just use case with a lowercase 'c'.

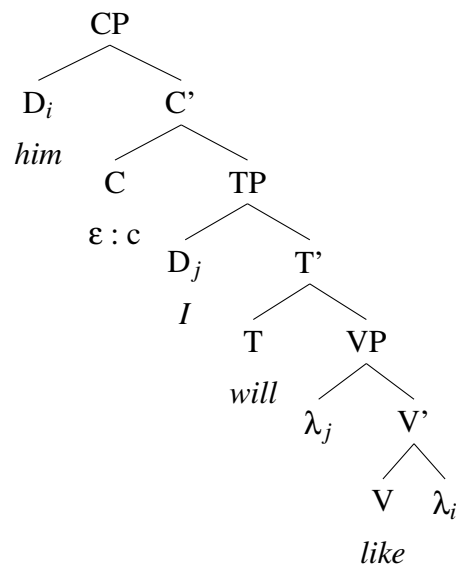


Figure 3.12

*fan* (meaning that something disastrous has occurred). However, if idiomaticity is indeed a diagnostic for constituenthood, then we again have the problem that, on the surface at least, the subject can appear separated from the verb and the object and yet the idiomatic meaning remains.

- (15) a. *The shit will hit the fan.*  
 b. *The shit must have really hit the fan.*  
 c. *The shit must have really been hitting the fan.*

However, if the subject is in fact base generated inside the VP, and if the correct generalisation is that idioms can only be formed from words which together form a constituent *at some point in the derivation*, then idiomaticity as a diagnostic for constituenthood holds. If idioms are stored as chunks of structure in the lexicon, which would explain their non-compositional interpretive properties, then we could go further and say that the words of an idiom must form a constituent in their base positions, their surface positions being irrelevant.

Another empirical argument for VPISH comes from floating quantifiers. Consider the following examples.

- (16) a. All those students have studied physics.  
 b. Those students have all studied physics.



The fact that 16a and 16b have identical interpretations suggests that just as the quantifier *all* clearly forms a constituent with the DP *those students* in 16a, so too does it in 16b *at a certain point in the derivation*. The basic idea (which originates with Sportiche (1988), a version of which is implemented in MGBank) is that *all the students* is first constructed as a QP constituent. This QP, like all arguments, is initially generated within the VP. Later on, either the whole QP will raise to the subject position, or the DP complement of the quantifier will break away and raise to the subject position on its own, thereby stranding the quantifier. The structures of 16a and 16b are therefore proposed to be along the lines of 17a and 17b respectively.

- (17) a. [<sub>TP</sub> [All those students]<sub>i</sub> have [<sub>VP</sub> t<sub>i</sub> studied physics]].  
 b. [<sub>TP</sub> [Those students]<sub>i</sub> have [<sub>VP</sub> [all t<sub>i</sub>] studied physics]].

Some evidence for this analysis of floating quantifiers (and against the alternative view that they are simply adverbials) comes from the fact that in many languages the quantifier and its antecedent must agree in number, gender and/or case whether or not the quantifier appears stranded from the DP. Number and gender agreement are exhibited in Portuguese, for instance (data from Hornstein et al. 2005, page 87).

- (18) a. Les filles ont toutes/\*tous dîné  
 the girls have all.FEM.PL/all.MASC.PL dined  
 ‘The girls had all had dinner’  
 b. Toutes/\*tous les filles ont dîné  
 all.FEM.PL/all.MASC.PL the.FEM.PL girls have dined  
 ‘All the girls have had lunch’

Case agreement between the floating quantifier and its DP antecedent is found in German.

- (19) Diesen Mädchen gefällt der Peter \*alle/allen  
 these.DAT girls pleases the.NOM Peter all.NOM/all.DAT  
 ‘These girls all like Peter.’

Of course, this analysis of floating quantifiers (which, as Hornstein et al. (2005) note, is not uncontroversial) relies on the assumption that all arguments of the verb are initially generated within the verb phrase, otherwise there would be no way to explain how the quantifier is left stranded immediately to the left of the verb. Floating quantifiers have therefore been argued to provide evidence for the veracity of VPISH.

A further, argument for VPISH comes from the fact that it allows for a much more transparent relation between syntactic and semantic argument structure than is otherwise possible. From very early on in TG, the subjects of passive and unaccusative verbs were analysed as being base generated within the VP as deep objects to the verb (owing to their status as THEME arguments) before moving to their surface subject positions. Consider the examples in 20 below.

- (20) a. Several problems will remain.  
 b. There will remain several problems.

In 20a, the DP *several problems* appears in the surface subject position (which we assume to be spec-TP) to the left of the auxiliary. However, it seems clear that this surface subject position is not a thematic one, owing to the fact that in 20b it is occupied by the expletive (i.e. meaningless) pronoun *there*. Interestingly, with this position occupied, the thematic DP shows up in the object position of the main verb. Transformationalists therefore standardly assume that the surface thematic subjects of unaccusative verbs such as *remain* start out their derivational lives in the object position of those verbs where they are assigned their THEME theta role. However, because unaccusative verbs by hypothesis lack the ability to assign/check accusative case, the object is forced to raise to the nearest position in which it can be assigned/check case, which in this instance is spec-TP of the same clause (to see how case is checked in 20b, see section 4.2.6 on Multiple Agree). The simplified structure of 20a is therefore assumed to be as in 21 below.

- (21) [<sub>TP</sub> [Several problems]<sub>i</sub> will [<sub>VP</sub> remain t<sub>i</sub>]]

A similar situation is found in the active-passive alternation seen in 24 below.

- (22) a. Jack solved the problems.  
 b. The problems were solved.

In 22a *the problems* appears in the object position, whereas in 22b *Jack* has vacated the subject position and the object is promoted to being a subject, much as we saw in 22b. In both 22a and 22b it plays the same thematic role. TG therefore assumes that passivizing a verb effectively transforms it into an unaccusative verb which lacks an external argument in spec-vP and is unable to assign accusative case, explaining why the object is forced to move to the subject position. This type of movement, which involves movement to an argument position, is referred to as A-movement within TG.

It contrasts with A'-movement which refers to movement to non-argument positions and includes the topicalisation movement we saw earlier as well as *wh*-movement and focus movement.

Returning to the motivation for VPISH, if both THEME arguments of unaccusatives and AGENT arguments of transitives are generated within the verb phrase, then we can say that *all* arguments of a verb are assigned their theta roles with that verb phrase, with only the structurally most prominent argument undergoing A-movement to the surface subject position. This in turn enables us to delineate the theta domain (VP) from the inflectional domain (TP), the latter being solely responsible for determining the tense/mood/aspectual properties of the predicate. From the perspective of the direct compositional approach to MG semantics of Kobele (2006), this simplifies the mapping between syntax and semantics because we can say that all theta roles are associated with the =d/d= selectee features located within the verb phrase.

We can in fact go further than this and tie specific theta roles to specific structural positions (an assumption embodied in Baker's (1988) Uniform Theta Assignment Hypothesis (UTAH)). Looking ahead slightly to the next section, we could say that a =d/d= on a main verb V is uniquely associated with a THEME theta role assigned to a DP, whereas =d on the null little *v* head is uniquely associated with the AGENT theta role. There is then no need for a grammar writer to write separate semantic types, as there is a one-to-one correspondence between syntactic and semantic types, the latter being entirely predictable from the former. This is not the case for many surface oriented formalisms, such as CCG for instance, where the same syntactic type may correspond to multiple semantic types (this point was made by Bos (2008, page 281)).

This transparency has the potential to make writing compositional semantics for MGs particularly quick and straightforward (of course, the work has to be done at some point, and writing MG syntactic grammars is correspondingly more onerous than writing CCG syntactic grammars). To better illustrate this point, consider the following examples.

- (23) a. Jack seemed to help.  
       b. Jack wanted to help.
- (24) a. Jack expected Pete to help.  
       b. Jack persuaded Pete to help.

Example 23a is superficially similar to example 23b. However, as discussed in more detail in section 3.3.6, this superficial similarity masks an important semantic

distinction: whereas *Jack* is clearly the AGENT of both *want* and *help* in 23b, he is only the AGENT of *help* in 23a, not of *seemed*. This is clear from the fact that *seem* can take an expletive subject, whereas *want* cannot.

- (25) a. It seemed that Jack helped.  
 b. \*It wanted that Jack helped.

A similar situation is found with 24a and 24b: whereas in 24b Pete is both a THEME argument of *persuade* and an AGENT argument of *help*, in 24a he is not a THEME argument of *expect*. This is again clear from the fact that only *expect* can take the expletive pronoun *there* as its object.

- (26) a. Jack expected there to be some help.  
 b. \*Jack persuaded there to be some help.

In CCGbank, raising-to-subject verbs like *seem* and subject control verbs like *want* are treated as having the same syntactic type:  $(S \setminus NP) / (S[to] \setminus NP)$ . Similarly, raising-to-object verbs like *expect* have the same syntactic type in CCGbank as object control verbs like *persuade*:  $((S \setminus NP) / (S[to] \setminus NP)) / NP$ . The distinction between these two pairs of verbs is therefore not made by the CCG syntactic types. Instead, it is made by the semantic types. What this means, however, is that additional and partially manual annotation must be performed at the semantic level for CCGbank, as indeed it has been (Bos, 2008).

In the MGbank grammar, on the other hand, *seem* and *want* have distinct syntactic types, as do *expect* and *persuade*. Moreover, their semantic types are entirely predictable from these syntactic types, and could thus be derived automatically. For example, consider the simplified MGbank syntactic types for *expect* and *persuade*.

*expect* :: t= v  
*persuade* :: c= =d v

For reasons which will not concern us here, object control verbs select CP clausal complements, whereas raising-to-object/ECM verbs select bare TP clausal complements in Minimalism. What is more immediately relevant here, is that only *persuade* has a =d feature to check. Assuming, as we said a moment ago, that =d on V checks a THEME/PATIENT theta role, it is easy to see that the distinction made in the syntactic types here will correspond in a very direct way with the required distinction in the semantic types. In MG as in CCG, each semantic type corresponds to a unique syntactic

type. However, in MG the relationship between syntax and semantics is arguably even more transparent, because only in MGs is this unique correspondence bidirectional, with every syntactic type also corresponding to exactly one semantic type.

### 3.2.8 The tripartite structure of clauses

Observe that in fig 3.12 the clause consists of three layers of projection: VP, TP and CP. In TG, these are taken to correspond to three very different types of semantics. VP is the domain of theta assignment; TP is the inflectional domain, determining properties such as tense, mood and aspect; and CP is the discourse domain, determining the illocutionary force of the clause and serving as the target of A'-movements such as topicalization, focus and wh-movement, all of which serve to render certain constituents more salient.

As we will see more detail in section 4.7, Minimalists assume that each of these three layers actually decomposes into multiple sub projections. For now, it will suffice to observe that the VP is standardly decomposed into an internal VP layer and a vP shell layer (Larson, 1988; Chomsky, 1995) headed by a light verb (known affectionately in the Minimalist literature as *little v*) which is null in English but in other languages turns up as an overt causative morpheme (see section 4.7.1). The AGENT subject is generated as the specifier of this little v head (before moving to spec-TP as before), while all other arguments are generated within the inner VP. Note that in MGbank this little v head uses the selectee *lv* because the upper- vs lowercase distinction is used for other purposes: uppercase licensors (e.g. +CASE) license overt movement while lowercase licensors (e.g. +case) license covert movement (both check and delete lowercase licensees (e.g. -case), i.e. upper- vs lowercase as no import on licensees; and uppercase D may persist after being checked to license control movement whereas lowercase d cannot - see sections 3.3.4 and 3.3.6. However, in the Xbar trees of this dissertation and in the MGbank corpus the projections of little v are still represented as v, v' and vP (rather than lv, lv', lvP) to make them more familiar to linguists.

A further point of note is that although until now we have simplified the discussion by assuming that only subjects check case, in fact objects are also assumed to check case somewhere within the vP via either overt or covert movement. In MGbank objects check case by overtly moving to spec-VP (following Chomsky 2008), with the verb then undergoing V-to-v head movement placing it to the left of all its complements (head movement is discussed in section 3.3.2). The derived Xbar structure of a simple

transitive sentence such as *he read the book* in MGBank is therefore as in fig 3.13.

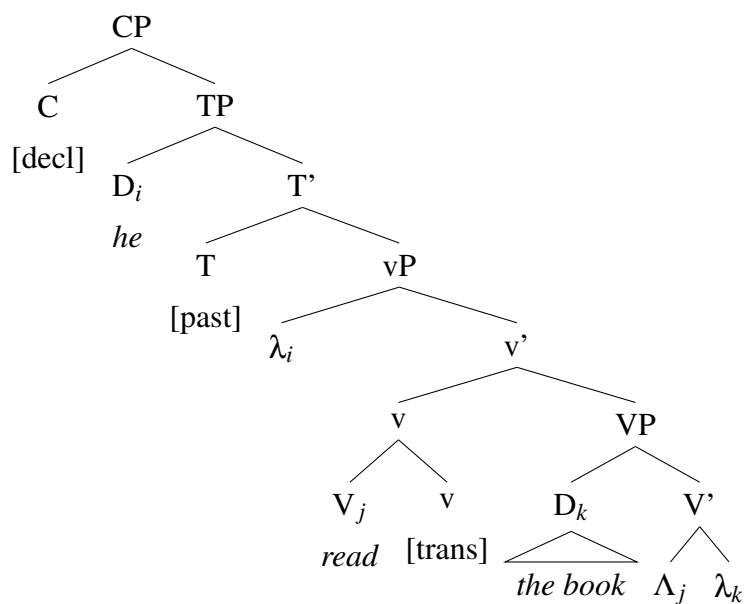


Figure 3.13: Derived Xbar structure for the transitive sentence *he read the book*.

### 3.2.9 Keeping null heads grounded

Null heads are very useful for encoding various types of linguistic generalizations. There is, however, some understandable skepticism outside of TG regarding the psychological plausibility of phonetically silent morphemes. One perceived problem is the fact that by equipping these heads with the relevant licensor features, they can be used in an ad hoc manner in order to derive any word order that is required. For this reason, one of the guiding principles that was adopted during the construction of the MGBank grammar was that null heads should, wherever possible, be semantically grounded in the sense that they encode some semantic property or other. This view contrasts with that of (Kayne, 1994, page 30), who argues that in addition to ‘contentful’ heads there are also heads which lack intrinsic content entirely and whose only purpose is to serve as targets for movement.

This project also takes the view that null heads should also ideally be phonetically grounded from a cross-linguistic perspective in the sense that they will ideally be overtly expressed as particles or bound morphemes in at least one of the world’s languages. For example, the null transitive/causative little *v* head which we saw in the previous section turns up as an overt bound causative morpheme in languages such

as Kannada (a Dravidian language spoken predominantly in India), as the following example (taken from Hornstein (2005, page 103) illustrates.

- (27) Naan-u neer-annu kud-is-d-e  
*I-NOM water-ACC boil-CAUS-PAST-1.S*

Null C clearly has overt versions in English (e.g. *that, for, if*), as does T, which in infinitivals is overtly realised as the particle *to*, and in certain (negative and interrogative) clauses as *do*, and the *-ed* past tense morpheme could be regarded as another overt instantiation of T in a full morphosyntactic system that includes affix hopping operations.

Aboh (1998) argues that the particles *ya* and *we* in Gungbe are plausibly analyzed as overt topic and focus heads respectively,<sup>17</sup> and even the null adjunctizers which are used to transform certain constituents into adjuncts in MGBank could plausibly be argued to turn up as *ly* suffixes on adverbs and on adjectives as the suffixes *ic, ist, al, y* etc. If the MGBank grammar were to be extended to handle morphology, many of the [adjunctizer] heads would be replaced with these overt morphemes, with head movement operations ensuring that they ended up being suffixed onto the items they select for. Similar in this respect are the [self] morphemes used for reflexive binding (see section 4.2.4), as these can be regarded as representing the *-self* morpheme in *himself, myself, themselves* etc.

All of this is not to claim that MGBank does not make use of ad hoc null heads at all. One example of an ungrounded head would be the [epp] heads which introduce +EPP licensors onto the various projections of auxiliary verbs in order to attract a floating quantifier to their specifier. For example, *all* may appear left adjacent to the main verb, the progressive auxiliary and the passive voice auxiliary, as illustrated in the examples below.

- (28) a. The men must have [<sub>vP</sub> all eaten lunch].  
 b. The men must have [<sub>progP</sub> all been eating lunch].  
 c. The men must have [<sub>voiceP</sub> all been eaten].

And in clauses with multiple auxiliaries, the floating quantifier exhibits some optionality in where it appears, and can even be left adjacent to the perfect auxiliary, as the following examples show.

- (29) a. The men must [<sub>perfP</sub> all have eaten lunch].

<sup>17</sup>See section 4.7.6 on the fine-grained structure of the CP domain.

- b. The men must have [<sub>vP</sub> all eaten lunch].

As we saw in section 3.2.7, there are good reasons to suppose that the floating quantifier *all* in these examples starts out its derivational life by forming a constituent with the DP *the men* (with the latter then breaking away and moving to spec-TP) rather than simply being an anaphoric adverb. The fact that the floating quantifier need not remain in the base generated spec-vP position, however, suggests that it too can undergo movement to a higher specifier position. The following null heads (simplified here for expository clarity) are therefore used in MGbank in order to transform each type of projection, vP, voiceP, progP or perfP, into a version of itself which attracts a floating quantifier (which has a -epp feature) to its specifier. In each case, the +FLOAT selectional requirement<sup>18</sup> ensures that only a floating quantifier (and not some other moving chain bearing a -epp licensee) is attracted to the relevant specifier position.

[epp] :: >lv= +EPP{+FLOAT} lv  
 [epp] :: >perf= +EPP{+FLOAT} perf  
 [epp] :: >prog= +EPP{+FLOAT} prog  
 [epp] :: >voice= +EPP{+FLOAT} prog

Such null heads are used in MGbank only as a last resort, however, and highlight obvious weaknesses in the theory presented here in its current form.

### 3.2.10 Derivational Economy and The Shortest Move Constraint

As we have seen, Minimalism attempts to reassign much of the work which was previously done by linguistic-internal UG Principles in the earlier GB theory to the requirements of other cognitive systems and to Chomsky's 'third factor' principles of efficient computation. One such principle is the condition of Last Resort, which states that 'computational operations must be driven by some condition on representations...to overcome a failure to meet such a condition' (Chomsky, 1995, page 28). The standard interpretation of this, at least in the case of Move, is that syntactic operations apply only when needed to check and delete some uninterpretable feature. As noted in Kobele (2006, page 20, fn 20), in MGs all syntactic features are uninterpretable in the sense that they must be eliminated (except for a single c selectee associated with the root of the tree), although some are associated with semantic features (following

<sup>18</sup>See section 3.4 on fine-grained selectional features.



Kobele (2006)) and hence could be described as interpretable in this sense. In MGs, both Move and Merge are always forced by the need to check and delete one or more syntactic features, hence MGs, including the one presented here, are in full accordance with Last Resort.

Another important economy condition which has featured prominently in one way or another in all iterations of Minimalism since Chomsky (1993) is the requirement that movement steps should be as short as possible. This general idea goes back most famously to Chomsky's (1964) A-over-A condition and Rizzi's (1990) Relativized Minimality. In Minimalism, it has been formulated variously as the Minimal Link Condition, the Attract Closest Principle, the Shortest Move Constraint, and, in more recent probe-goal theory, in terms of Minimal Search; Chomsky and others now often simply uses the informal term *intervention effects*. Here we will refer to the Shortest Move Constraint and abbreviate this as CSMC when talking about Chomsky's (1993) version. This is done in order to differentiate it from the stricter version introduced by Stabler (1997) for MGs and abbreviated here as elsewhere in the literature as SMC. To understand the basic idea behind CSMC, consider the configuration shown below.

(30)                                    [ ... A ... [ ... B ... [ ... C ... ] ] ]

There are two general cases to consider. In the first case, C is a mover and both A and B are potentially viable targets of the movement. CSMC requires that C not skip B by proceeding directly to A, although it is perfectly possible for C to move successive cyclically first to B and then to A. The sort of phenomena which are captured by this aspect of CSMC include those which in earlier theory were captured by Rizzi's (1990) Relativized Minimality condition. One well-known example is the arguably universal ban on so-called superraising across the world's languages. Consider the following contrasts for instance.

- (31) a. Jack seems to be likely to win.  
       b. It seems that Jack is likely to win.  
       c. \*Jack seems that is likely to win.  
       d. \*Jack seems that it is likely to win.

31a and 31b mean (virtually) the same thing. In either case, in the theory being developed here, *Jack* starts out in spec-vP of the *win* clause and moves to a higher position: spec-TP of the intermediate *be* clause in 31b and spec-TP of the matrix *seems*

clause in 31a. 31c shows that if the intermediate clause is finite, it is not possible for *Jack* to move all the way to the matrix spec-TP position as he did in 31a where the intermediate clause is non-finite; instead, *Jack* must move to spec-TP of the intermediate finite clause where he is frozen in place.

31d is the superraising case, and shows that it is not possible to circumvent this freezing restriction on *Jack* by using a dummy expletive pronoun to fill the intermediate finite spec TP position (and check the EPP/case features of the intermediate T). CSMC provides an explanation for this fact: the intermediate spec-TP position is a *potential* landing site for *Jack*. The fact that *Jack* cannot actually land there owing to the expletive filling this position means that the Shortest Move Constraint cannot be satisfied as *Jack* would have to skip a potential landing site in order to arrive at the matrix spec-TP position. In 31a, meanwhile, either *Jack* proceeds to the matrix spec-TP successive cyclically via the intermediate spec-TP (but is not frozen in place either because non-finite T is deficient in features as in Chomsky (2001b), or simply does not attract a specifier at all as in Epstein and Seely (2006) - see section 4.7.3).

Another phenomenon which this first aspect of CSMC has been used to account for is the ban on wh-island violations. Consider the following contrast.

- (32) a. Who<sub>i</sub> did Jack say that Mary loves t<sub>i</sub> ?  
 b. Jack wondered who<sub>i</sub> Mary loves t<sub>i</sub>.  
 c. \*Who<sub>i</sub> did Jack wonder [why<sub>j</sub> Mary loves t<sub>i</sub> t<sub>j</sub>] ?  
 d. \*Why<sub>i</sub> did Jack wonder [who<sub>j</sub> Mary loves t<sub>j</sub> t<sub>i</sub>] ?

In example 32a, the wh word is extracted across a clause boundary to the left periphery of the sentence without problem. In 32b the same wh word is only extracted as far as the left periphery of the embedded clause and the result is again a perfectly acceptable sentence. Examples 32c and 32d, however, attempt to combine these two possibilities by introducing a second wh word and attempting to move one to the embedded clause left periphery and the other to the matrix clause left periphery. Whether or not it is the argument *who* or the adjunct *why* which attempts to move to into the matrix clause, the result is an ungrammatical sentence. CSMC provides an account for this because, in either case, the item moving into the matrix clause would have to pass over the embedded left peripheral position which is already filled by the other wh word, thereby skipping a potential landing site.

The second basic scenario which the Shortest Move Constraint is intended to capture is where both B and C in 30 are moving to check the same licensee feature and A

is a potential target of movement for both. In this case, because B is structurally closer to A than C is (B being less deeply embedded than C), only B is permitted to move. This aspect of CSMC is intended to account for phenomena which were traditionally handled by Chomsky's 1973 Superiority Condition. Consider the following examples taken from (Chomsky, 1993, page 14).

- (33) a. Whom<sub>i</sub> did John persuade t<sub>i</sub> [to visit whom<sub>j</sub>].  
 b. \*Whom<sub>j</sub> did John persuade whom<sub>i</sub> [to visit t<sub>j</sub>].

As the above contrast indicates, in English, when there is more than one *wh* constituent present in a single clause, only the highest is permitted to move to its left periphery. Chomsky (1993) argues that 33b should be barred by the Shortest Move Constraint: the movement of the higher *whom* to the left periphery of the sentence is clearly shorter than, and hence should block, the movement of the lower one.

Although the Shortest Move Constraint is an intuitive idea, (Chomsky, 1993, page 15) concedes that 'spelling out these notions to account for the range of relevant cases is not a trivial matter,' and, as noted, there have been a number of implementations of this idea in MP. In current Agree-based Minimalism, CSMC is implemented in terms of Minimal Search of a probe (the head bearing the licenser feature) into its complement domain. This search proceeds top down meaning that a structurally superior goal will be located before a more deeply embedded one.

In MGs as we have seen, phrase structure trees are collapsed and there is thus no notion of relative structural distance of different goals from a given probe. In Stabler (1997) a much stricter version of the Shortest Move Constraint is proposed which has been adopted by all subsequent work in the MG framework, including the formalism presented here. This version is given repeated below from section 3.2.6 and will be referred to throughout as SMC.

**The Shortest Move Constraint (SMC):** Two licensee features may both be active at the same time in the derivation only if they are distinct.

From the perspective of weak generative capacity, this version of the Shortest Move Constraint is crucial for achieving MCFG-equivalence: Salvati (2011) shows that 'the membership problem of minimalist grammars without the shortest move constraint is as difficult as provability in Multiplicative Exponential Linear Logic'. From a linguistic/strong generative capacity perspective, SMC enforces various locality constraints

including *wh*-islands and the ban on superraising discussed above. Consider again the ungrammatical superraising example 31d, repeated as 34 below.

(34) \*Jack seems that it is likely to win.

As discussed in more detail in section 4.7.2, in Minimalism all (overt and argumental) DPs are standardly assumed to have case features to check; in early MP and in most MGs, including here, this is done via movement. In order to derive this sentence, therefore, the parser would need to create the following intermediate expression for the stage of the derivation immediately preceding the checking and deletion of the +CASE licensor on the T hosting *is*.

is likely to win : +CASE t, it : -case, Jack : -case

Both *it* and *Jack* have the same active licensee feature, hence this expression will be disregarded by the parser owing to SMC meaning that the final superraising sentence is blocked. The *wh*-island violation cases are barred in precisely the same way. At a certain point in the derivation of either 32c or 32d, the parser would need to construct the following VP expression which violates the SMC by having more than one moving chain with the same active -*wh* licensee.

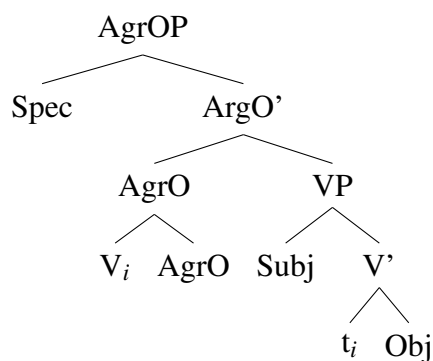
loves : v, who : -wh, why : -wh

It is important to recognise that Stabler's version of the SMC is much more restrictive than its MP counterpart (whichever implementation one considers). For example, as well as correctly blocking the superiority violation example 33b, it also incorrectly blocks 33a owing to the fact that the parser would in this instance need to allow for two -*wh* features to be simultaneously active in the derivation (on the standard assumption that the rightmost *wh* word moves covertly to the left periphery just as the leftmost one moves there overtly<sup>19</sup>). One could of course split *wh* into separate *wh<sub>who</sub>* and *wh<sub>why</sub>* licensor/licensees to avoid the SMC violation here. However, as well as missing a fundamental generalisation (both movements are an instance of a single *wh*-movement type), this would also mean that superiority effects and *wh*-islands were not enforced. At present the MGbank formalism/grammar is in fact unable to generate multiple-*wh* examples such as 33a owing to SMC. However, see Gärtner and Michaelis (2010) for

<sup>19</sup>In multiple *wh*-fronting languages, such as Bulgarian, all *wh* constituents move overtly, so even if we tried to claim that in English the second *wh* item does not move, this would not help us for several other languages.

a wh-clustering approach to multiple wh fronting which solves this problem and could be implemented in future for MGBank.

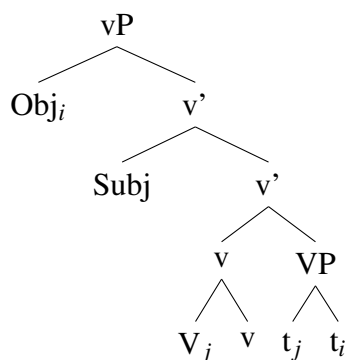
Before closing this section, it is worth noting that the formalism presented here does not make use of any notion of a *Minimal Domain* which was crucial in Chomsky's (1993) formulation of the Shortest Move Constraint. Chomsky assumed that both subjects and objects are initially generated within the VP and then raise to higher functional projections to check agreement and case features. In the case of objects, these move to the specifier of AgrOP (covertly in English), which immediately dominates VP. The stage in the derivation at which the AgrOP has been constructed is shown below.



Subsequently, on the LF cycle, the object will move to spec-AgrO in apparent violation of the SMC given that the subject is base generated in a structurally higher position than the object. Chomsky (1993) argues that in fact the two arguments are *Equidistant* to spec-AgrO because they are contained in the same Minimal Domain, where a Minimal Domain of a head H is defined as the set of categories immediately contained within projections of H. Both Subj and Obj in the above are therefore in the Minimal Domain of V and therefore Equidistant from spec-AgrO as far as the SMC is concerned, which allows the object to cross the subject (and the subject to cross the object when it later moves to spec-AgrS).

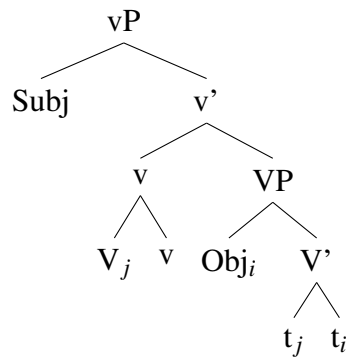
Under Stabler's strict version of the SMC, however, the creation of crossing (or nested) dependencies which check the same licensee features (here case/agreement features) are impossible, and there is no concept of Equidistance or Minimal Domains. Instead, the grammar must be designed in such a way that the object first checks its case/agreement features *before* the introduction of the subject into the derivation, thereby avoiding the need for crossing dependencies. In (Chomsky, 1995, page 352) AgrO was eliminated and VP was split into two phrases (following Larson (1988)): an inner VP core and an outer vP shell headed by a null light verb. The object is generated

within the VP core while the subject is generated as the inner specifier of the vP. The object then raises to become the outer specifier of vP to check case and agreement features. The stage in the derivation at which the vP has been constructed is shown below (Note that some authors regard object movement to spec vP as proceeding covertly in English, while others regard it as proceeding overtly. Here, it is shown as overt movement).

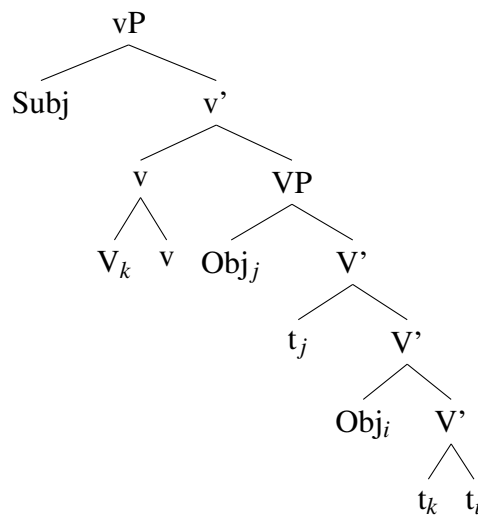


In this revised system, the subject and object are generated in different Minimal Domains ( $\min(v)$  and  $\min(V)$  respectively); however, Equidistance also applies to the *targets* as well as the *sources* of movement, and because the subject and the target of the object's movement are both in  $\min(v)$ , they are Equidistant from the object's base position and so the crossing dependency can be established without problem.

Although we have seen that the MGBank grammar adopts the vP-VP shell analysis of verb phrases, this will not help us with the problem that crossing dependencies formed through the checking of the same licensee features (here -case) are banned under Stabler's strict SMC. To avoid this problem, the MGBank grammar adopts a suggestion made in Chomsky (2008) that in fact the object moves to spec-VP in order to check case (Chomsky assumes V to have inherited the relevant licenser features from v just as he assumes T to inherit them from C). Because the object checks its case feature before the point at which the subject is introduced into the derivation, there will be no SMC violation triggered. The stage at which the vP has been constructed according to the MGBank analysis is shown below.



Where there is more than one internal DP argument (i.e. in double object constructions such as *he gave the man the suitcase*), the rightmost object is introduced as the complement of the verb and has its case feature checked by moving to spec VP, after which the second object is generated as a second specifier of the main verb before moving to a third, outermost spec VP position to check its -case feature. In this way, there are never any crossing or nested dependencies as far as case checking is concerned and SMC is always respected. The stage of the derivation at which the vP of a double object verb has been generated in MGBank is shown below.



### 3.3 Extended Directional Minimalist Grammars

Section 3.2 presented the core DMG formalism. This section provides the details of a number of extensions which were incorporated into this core formalism in order to allow it to more adequately describe a much wider range of linguistic phenomena. We will refer to the resulting formalism as Extended Directional Minimalist Grammars (EDMGs).

### 3.3.1 Adjunction

(Non-Kaynian) Minimalists standardly assume that in addition to complements and specifiers, a third type of *adjunct* dependent can be formally distinguished. Adjuncts are usually (though not exclusively) semantically adverbial, and include adverbs themselves as well as modificational PPs, adjectives and arguably at least some relative clauses. They also display a number of defining properties, such as iterativity, a high degree of optionality, and type-preservation. These properties are illustrated by the following examples.

- (35) a. The student.  
 b. The [<sub>NP</sub> [<sub>NP</sub> student ] [<sub>PP</sub> with the red hair]].  
 c. The [<sub>NP</sub> [<sub>NP</sub> [<sub>NP</sub> student ] [<sub>PP</sub> with the red hair]] [<sub>PP</sub> from Cambridge]].  
 d. The [<sub>NP</sub> [<sub>NP</sub> [<sub>NP</sub> [<sub>NP</sub> student ] [<sub>PP</sub> with the red hair]] [<sub>PP</sub> from Cambridge]] [<sub>PP</sub> in the leather jacket]].

Example 35a shows that the NP *student* need not take any PP modifiers; however 35b-d show that it is possible to optionally and iteratively adjoin PPs to this NP. This iterativity property differentiates adjunct PPs from complement PPs, as a noun can only take one complement, as illustrated below.

- (36) a. The student of physics.  
 b. \*The student of physics of chemistry.

The unacceptability of 36b indicates that *the student* in this example is not of the same type as *the student of physics*; if it were, then it should be possible for *the student of physics* to take *of chemistry* as a complement. Thus complementation, unlike adjunction, is not type-preserving.<sup>20</sup> Moreover, if a complement PP is present, it must be the closest PP to the head.

- (37) a. The student of physics with the red hair from Cambridge.  
 b. \*The student with the red hair of physics from Cambridge.

Adjunct PPs adjoining to the same phrase, on the other hand, may transpose freely with one another. Compare 35d with 38.

<sup>20</sup>In the case of NPs, complements are, like adjuncts, often highly optional. However, in the verbal domain this is not always the case. For example, the verb *devour* obligatorily takes a complement DP, but may take zero or more adjuncts (*he devoured \*(his food) (hastily) (without any cutlery) (yesterday morning)*). This indicates that whereas governing heads select for their complements, they do not (usually) select for their adjuncts. Instead, it is adjuncts which seem to select for their governors.



(38) The student in the leather jacket with the read hair from Cambridge.

These facts indicate that, unlike complementation, each adjunction results in a larger constituent which preserves the type (here NP) of the constituent adjoined to.

It should be noted at this point that this discussion abstracts away from certain messy details one encounters when annotating realistic corpora. For example, it is sometimes quite hard to distinguish adjunct PPs from complement PPs, and there is clearly inter-speaker and intra-speaker variation with respect to how these items are analysed. This is reflected in the fact that the PTB uses a tag CLR (closely related) to indicate marginal cases. The fuzziness of the complement/adjunct distinction in certain cases does not invalidate its formal status or its overall usefulness, however, and we will assume that the distinction is absolute from a formal perspective here.

MGParse incorporates the approach to leftward adjunction of Frey and Gärtner (F&G) (2002) and extends it with rightward adjunction. F&G introduce a new category into the lexicon, which we will refer to here as an *adjunctizer*, along with a new adjunction selector diacritic  $\approx$ . Adjunctizers are effectively unary functions which transform non-adjunct types into adjunct types. In MGbank, they can be null or overt. For example, shown below are two adjunctizers which map CPs and PPs respectively into rightward adjuncts that adjoin to VPs.

because :: c =  $\approx$ v

[adjunctizer] :: p =  $\approx$ v

We must also add the ADJOIN rules in fig 3.14 to the grammar to accommodate the new  $\approx$ x/x $\approx$  adjunction selector features. Notice that these rules involve *asymmetric* checking in contrast to the earlier MERGE rules: only the selector feature, not the selectee feature, is deleted; this captures the optional, iterative and type-preserving properties of adjunction. Note also that this time the head features of the mother derive from the selectee not the selector, making the selectee the head.

As in the standard Merge rules, these Adjoin rules are divided into those in which the dependent will begin to move following the adjunction (Adjoin3 and Adjoin4) and those in which it will not (Adjoin1 and Adjoin2); they are also divided according to whether the dependent adjoins to the left (Adjoin1 and Adjoin3) or to the right (Adjoin2 and Adjoin4) of the governor. Notice that in these rules, the adjunct is represented as not having any movers inside it. In other words, these rules implement a strict version of the Adjunct Island Constraint (Huang, 1982). In actual fact, the adjunct island

$$\frac{[t; x \approx]}{[ts : x\gamma, \alpha_1, \dots, \alpha_k]} (ad\ join1)$$

$$\frac{[s; x\gamma, \alpha_1, \dots, \alpha_k] \quad [t; \approx x]}{[st : x\gamma, \alpha_1, \dots, \alpha_k]} (ad\ join2)$$

$$\frac{[t; x \approx \delta] \quad [s; x\gamma, \alpha_1, \dots, \alpha_k]}{[s : x\gamma, t : \delta, \alpha_1, \dots, \alpha_k]} (ad\ join3)$$

$$\frac{[s; x\gamma, \alpha_1, \dots, \alpha_k] \quad [t; \approx x \delta]}{[s : x\gamma, t : \delta, \alpha_1, \dots, \alpha_k]} (ad\ join4)$$

Figure 3.14: Sub-functions of ADJOIN

constraint arguably only applies to clausal adjuncts. For example, it is often possible to extract the DP complement of an adjunctival preposition, thereby stranding said preposition. This is illustrated by the following dialogue.

SPEAKER 1: I left the house [<sub>PP</sub> without my jacket].

SPEAKER 2: What<sub>i</sub> did you leave the house [<sub>PP</sub> without t<sub>i</sub>]?

For this reason, the MGBank formalism does in fact allow non-clausal adjuncts to contain movers. See section 4.3.1.2 for further details and illustrative examples.

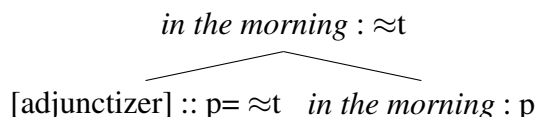
To see how the rules in fig 3.14 work in practice, assume that we have already constructed the TP *he drove her to work* whose collapsed chain is given below, and that we now wish to right adjoin the temporal PP *in the morning*, also shown in collapsed tree format below, to this TP. We can achieve this using the null adjunctizer also shown below in our lexicon.

he drove her to work : t

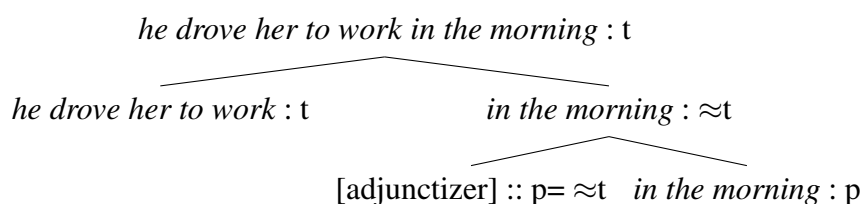
in the morning : p

[adjunctizer] :: p =  $\approx$ t

First, we merge the adjunctizer with the PP using Merge1. This results in the collapsed tree shown at the root of the following abbreviated derivation tree.



Notice that the null string of the adjunctizer is only represented as [adjunctizer] at the lexical level, and does not appear at all at the root node. This is the convention used in the MGbank trees. Because the operation just represented was a case of standard Merge, both the selector  $p =$  and selectee  $p$  features were deleted. Our new constituent has the rightward adjoin feature  $\approx t$  as its first feature and the TP *he drove her to work* has  $t$  as its first feature. We can therefore apply the rule Adjoin2 to merge these two items into the larger TP shown at the root of the new derivation tree below.



In this case, only the selector  $\approx t$  is deleted; the  $t$  selectee remains intact in accordance with the type preserving property of adjunction. It is worth noting that this approach precludes the possibility of the classical analysis of adjuncts in early GB as daughters and sisters to  $X'$ . This is because adjunction can only take place once the selectee feature of a phrase is exposed, at which point it is already an XP. In order to enable adjunction to  $X'$  we would need to allow for selectors of selectors and selectors of licensors. For instance, assume that instead of adjoining the adjunct to the TP *he drove her to work* we wanted to adjoin it to the the  $T'$  *drove her to work*, shown in collapsed tree format below.

drove her to work : +CASE  $t$ , he : -case

Once the +CASE feature is checked and deleted by moving *he* to spec-TP this constituent will cease to be an intermediate  $T'$  and will become a maximal TP. We therefore need to adjoin the adjunct before that happens. We could in principle achieve this using a feature such as  $\approx(+CASE)$  on the adjunct, which would check but not delete the +CASE licensors of a constituent to its left. In other cases we would need to select for a selector. For example, consider the following hypothetical  $V'$  category.

helped her : =d  $v$

This expression already contains a DP complement and is now looking for another DP to its left. Once it selects that DP it will no longer be a V' and will become a VP. To right adjoin to V' we would therefore need to use a feature like  $\approx(=d)$ . MGBank bans such second order selectional features and adjunction is strictly only to XP (which is in accordance with current non-Kaynian Minimalist analyses). This restricts the distribution of adjuncts and allows for a simpler, more elegant feature system than would otherwise be the case.

Finally, it is worth noting that there are strong (arguably universal) constraints on the ordering of certain types of adjuncts. To account for this, Cinque (1999) argues that there is a universal hierarchy of some 30 functional heads located along the spine of the clause, each of which may select for a different type of adverb in its specifier (Cinque adopts Kayne's LCA and therefore treats adjuncts and specifiers as being configurationally identical). These functional heads are strictly ordered, which provides an explanation for contrasts such as the following.

- (39) a. *Frankly, Jack probably once usually arrived late.*  
 b. *\*Usually, Jack late frankly once arrived probably.*

Similar restrictions on ordering are also found in the nominal domain with respect to adjectival adjuncts.

Constraints on adjunct ordering are only represented at a very coarse level of granularity in MGBank. For example, there are discourse adjuncts which usually adjoin to CP, temporal adjuncts which usually adjoin to TP, manner adjuncts which usually adjoin to VP, and so on. In practice, however, even here it was often necessary to allow exceptions. For instance, in the following example from MGBank, the PP *with some charities* has undergone topicalisation to spec-CP; because the specifiers of a projection are necessarily inside of its adjuncts in the present formalism, this means that the discourse adjunct *however* in this example cannot be adjoined to CP, which is the domain of discourse (see section 4.7.6).

- (40) *With some other charities, however, it's the other way around.*

Instead, *however* in this example adjoins in MGBank to TP, which is the inflectional domain of tense/mood/aspect etc, placing it to the right of the topicalised constituent. An alternative to this would be to decompose CP into multiple projections of the type proposed in Rizzi (1997) (see section 4.7.6) and adjoin *however* to an inner one of these and then move the topicalised phrase to an outer head, such as topP.

It would be perfectly possible to fully implement Cinque's proposals using the MG formalism, of course, but doing so would involve allowing for many more null heads along the clause of the spine than are currently used in this grammar. There is then the question of whether to include all 30 heads in *every* tree, regardless of whether or not the adverbial it is supposed to support is actually present. However, including all of these heads leads to extremely large trees which are difficult to read and contain many semantically inert heads. On the other hand, including them only for the cases in which they are needed leads to an explosion in the size of the lexicon as there must be separate versions of each head to ensure that it can select for every other head that may appear below it (see chapter 7 of Fowlie (2015) for critical discussion of this approach).

An interesting alternative to adjunct ordering in an MG context which does not require the postulation of additional abstract null heads can be found in Fowlie (2013). Here, the negative polarity version of each *sel* feature is modelled as  $[x, y]$  (rather than just  $x$ ), where the  $x$  encodes the basic selectee category and the  $y$  encodes the last modifier type (if any) which adjoined to it. Fowlie then imposes a partial ordering on the set *sel* and introduces an operation *Adjoin* which ensures that an adjunct of category  $[z, z]$  can only adjoin to a category  $[x, y]$  if  $z \geq y$  in the hierarchy. Adapting the MGbank formalism so as to include this approach to adjunct ordering is left for future research.

### 3.3.2 Head Movement

All instances of movement which we have so far considered have been phrasal in the sense that they involved the movement of some maximal XP level projection. However, a standard (but by no means uncontroversial) assumption within TG since the mid- to late 1970s has been that languages also feature so-called *head movement* operations involving the movement of just the X head of a phrase to adjoin to the head which governs it (see, e.g. Emonds (1978)).

The vanilla case of head movement in English is found in main clause interrogatives, such as 41 below, in which the finite auxiliary appears in the pre-subject position which in embedded interrogatives, such as 42, is occupied by an overt complementizer.

(41) Has he seen you?

(42) I want to know [if he has seen you]

It is not possible for *both* the complementizer and the finite auxiliary to appear in the clause-initial position, as example 43 below illustrates.

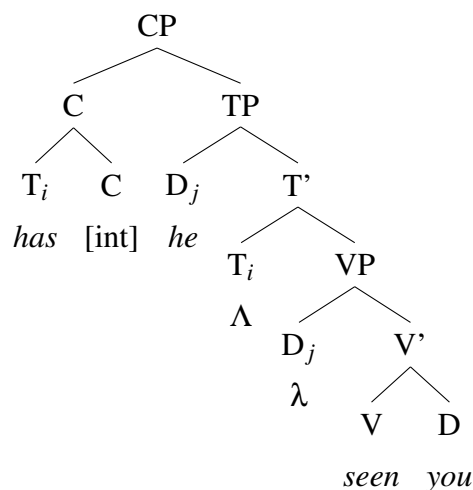


Figure 3.15

(43) \*I want to know if has he seen you.

One explanation for the complementary distribution of complementizers and finite auxiliaries in English interrogatives is that these items both appear in the same position, namely C. However, Minimalists standardly assume that there is a fixed hierarchy of heads in the clause, with temporal heads appearing below complementizer heads. Given that even in its inverted position the above auxiliary still clearly encodes both perfect aspect (because the root lexeme is *have* and the main verb is able to take the -en suffix) and present tense (because of the -s suffix attached to *have*) it does not seem plausible in the context of the current framework to suppose that the finite auxiliary is base-generated in the C position.

MGBank therefore adopts the standard TG view that the auxiliary starts out in T but then undergoes T-to-C head movement in main clause questions, adjoining to the left of a null interrogative C head morpheme. The (simplified) structure used in MGBank for embedded *yes-no* interrogatives such example 41 above is shown in fig 3.15;  $\Lambda$  is used here and in the MGBank derived/Xbar trees to indicate a trace of head movement; it contrasts with  $\lambda$  which is used for (leftward) phrasal movement.

Stabler (2001b) shows how head movement can be incorporated directly into the Merge rules of an MG without extending the expressive power of the formalism. His key insight is that the lexical head string of an expression must be kept separate from its left and right dependent strings until that expression has itself been merged/adjoined as a dependent, in case the head string subsequently has to undergo head movement. Stabler then introduces new feature diacritics which are added to certain selector features

and trigger head movement with head adjunction either to the left ( $>$ ) or the right ( $<$ ) of the governing head (the latter option is used in certain phrasal verb constructions in MGBank). Fig 3.16 gives the MERGE rules for rightward selection with leftward or rightward adjoining head movement, where the  $\varepsilon$  indicates an empty string (leftward selection with accompanying head movement is also permitted by the formalism, but in practice was not required for the MGBank corpus owing to the fact that complements almost always follow their heads in English).

$$\frac{[\varepsilon, s_h, \varepsilon :: >x = \gamma] \quad [t_l, t_h, t_r; x, \alpha_1, \dots, \alpha_k]}{[\varepsilon, t_h s_h, t_l t_r; \gamma, \alpha_1, \dots, \alpha_k]} (\text{merge\_hm1})$$

$$\frac{[\varepsilon, s_h, \varepsilon :: >x = \gamma] \quad [t_l, t_h, t_r; x \delta, \alpha_1, \dots, \alpha_k]}{[\varepsilon, t_h s_h, \varepsilon; \gamma, t_l t_r; \delta, \alpha_1, \dots, \alpha_k]} (\text{merge\_hm2})$$

$$\frac{[\varepsilon, s_h, \varepsilon :: x <= \gamma] \quad [t_l, t_h, t_r; x, \alpha_1, \dots, \alpha_k]}{[\varepsilon, s_h t_h, t_l t_r; \gamma, \alpha_1, \dots, \alpha_k]} (\text{merge\_hm3})$$

$$\frac{[\varepsilon, s_h, \varepsilon :: x <= \gamma] \quad [t_l, t_h, t_r; x \delta, \alpha_1, \dots, \alpha_k]}{[\varepsilon, s_h t_h, \varepsilon; \gamma, t_l t_r; \delta, \alpha_1, \dots, \alpha_k]} (\text{merge\_hm4})$$

Figure 3.16: Head Movement functions

As an example of how these rules work, consider the following two expressions for a yes-no interrogative C head and TP constituent respectively.

$\varepsilon, [\text{int}], \varepsilon :: >t = c$   
 he, did, drive her to work : t

The interrogative C head is looking for a constituent with a t selectee as its first feature and will trigger head movement of the head string of that TP constituent to the left of its own head string owing to the  $>$  diacritic. Merging the above two expressions therefore results in the collapsed tree at the root of the following derivation tree.

$\varepsilon, \text{did} [\text{int}], \text{he drive her to work} : c$   
 $\varepsilon, [\text{int}], \varepsilon :: >t = c \quad \text{he, did, drive her to work} : t$

In order to ensure that heads are always accessible for head movement, all of the Merge, Move and Adjoin rules from sections 3.2.6 and 3.3.1 must be formulated so that

the head string of every head chain is kept separate from its left and right dependents. Once a head chain's string is either fused into the complement string position of a governing head, or becomes the string of a moving chain, there is no longer any reason to keep its head string component separate from its left and right dependent string components because it is no longer susceptible to head movement at this point. The revised versions of the Merge, Move and Adjoin rules from sections 3.2.6 and 3.3.1 which keep the head string separate from its left and right dependents can be found in Appendix B.

### 3.3.3 Rightward Movement

Objects must canonically appear immediately to the right of their governing verbs in English, as the following contrast illustrates.

- (44) a. She introduced him to the guests.  
 b. \*She introduced to the guests him.

As is well known, however, if the DP is phonetically 'heavy', it may undergo so-called heavy NP shift (or, under the DP hypothesis (Abney, 1987) adopted here, heavy DP shift) to the end of the sentence, as illustrated by the following.

- (45) She introduced  $t_i$  to the guests [the famous detective from Belgium] $_i$

Rightward movement can also target other categories besides DPs, including PPs (46 and 47), and CPs (48, 49 and 50).

- (46) a. I read a passage from McEwan's latest novel yesterday.  
 b. I read a passage  $t_i$  yesterday [ $_{PP}$  from McEwan's latest novel] $_i$ .
- (47) a. Mr Rapanelli met with U.S. Assistant Treasury Secretary David Mulford in August.  
 b. Mr Rapanelli met  $t_i$  in August [ $_{PP}$  with U.S. Assistant Treasury Secretary David Mulford] $_i$ .
- (48) a. They will report the fact that they failed totally and utterly tomorrow.  
 b. They will report [ $_{DP}$  the fact  $t_i$ ] tomorrow [ $_{CP}$  that they failed totally and utterly] $_i$ .
- (49) a. That their meeting never took place is a shame.  
 b. [ $_{DP}$  It  $t_i$ ] is a shame [ $_{CP}$  that their meeting never took place] $_i$ .



- (50) a. I called a friend who I hadn't spoken to in ages yesterday.  
 b. I called a friend  $t_i$  yesterday [ $_{CP}$  who I hadn't spoken to in ages] $_i$ .

Traditionally these types of constructions were analysed in TG as involving rightward movement. However, Kayne's (1994) highly influential Linear Correspondence Axiom entails that rightward movement (like rightward adjunction, leftward complementation and rightward specification) does not exist, and this has resulted in rightward movement being largely abandoned in Minimalism.<sup>21</sup> However, this often necessitates using additional (and often semantically unmotivated) silent heads to trigger elaborate sequences of multiple (remnant) leftward movements for what is intuitively a single weight-theoretic requirement: that a heavy constituent appear sentence-finally. And there is a practical consideration here as well: the Penn Treebank (PTB) uses rightward movement to capture constructions such as right node raising and certain constructions featuring expletive *it* (such as 49 above); retaining this operation therefore allows the MG trees to remain closer to their PTB counterparts, which in turn facilitates the dependency mapping-based scoring method that is used during the automatic phase of the treebank generation process described in the next chapter. MGbank therefore follows the minority of Minimalists who retain rightward movement in the grammar.

Rightward movement is standardly assumed to be to adjoined positions: like standard adjunction, it is type preserving (meaning that here it will involve asymmetric feature checking) and is most plausibly triggered by a requirement of the dependent (i.e. the mover) rather than the governor (i.e. the target of the movement). Here we will adopt the suggestion in Gärtner and Michaelis (2003) to treat rightward movement as a rightward version of the leftward scrambling operation introduced in Frey and Gärtner (2002). Their approach makes use of a  $\sim x$  leftward scrambling licensee, and here we will represent its rightward counterpart as  $x\sim$ . Selectee  $x$  features will now serve a second purpose as licensors for rightward movement. The rightward movement rule is given in fig 3.17. Note that as far as all the other Merge, Move and Adjoin rules presented in this dissertation are concerned,  $x\sim$  has precisely the same effect as any other type of licensee feature: it will cause an expression/chain that is Merged/Adjoined/Moved to (keep) moving.

$x\sim$  features enter the derivation on null [extraposer] heads which map items into rightward-moving versions of themselves. As an example of how this works, we will look at the derivation of the VP *introduced to the guests the famous detective from*

---

<sup>21</sup>Ernst 2002 is a notable exception.

$$\frac{[s_l, s_h, s_r : x\gamma, \alpha_1, \dots, \alpha_{i-1}, t : x\sim, \alpha_{i+1}, \dots, \alpha_k]}{[s_l, s_h, s_r t : x\gamma, \alpha_1, \dots, \alpha_{i-1}, \alpha_{i+1}, \dots, \alpha_k]} r\_move$$

Figure 3.17: The rule for rightward movement

*Belgium* from example 45. Assume that the derivation has reached the point where the workspace includes the following lexical and derived expressions.

to the guests : p  
 the famous detective from Belgium : d -case  
 introduced :: p= =d +CASE v  
 [extraposer] :: d= +CASE d -case v~

First, we merge the verb with its PP complement, resulting in the V' expression shown in collapsed tree format at the root of the following derivation tree.

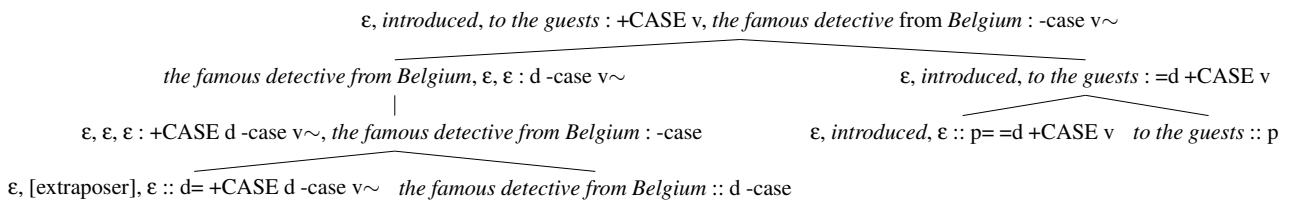
$$\begin{array}{c} \varepsilon, \textit{introduced}, \textit{to the guests} : =d +CASE v \\ \hline \varepsilon, \textit{introduced}, \varepsilon :: p= =d +CASE v \quad \textit{to the guests} :: p \end{array}$$

Next, we merge the heavy DP *the famous detective from Belgium* with the null [extraposer] head, and then move the former to check -case against the +CASE feature of the latter. Effectively, the [extraposer] ‘pied-pipes’ the d and -case features of the object, replacing these with its own.<sup>22</sup> This results in the new DP shown at the root of the following derivation tree.

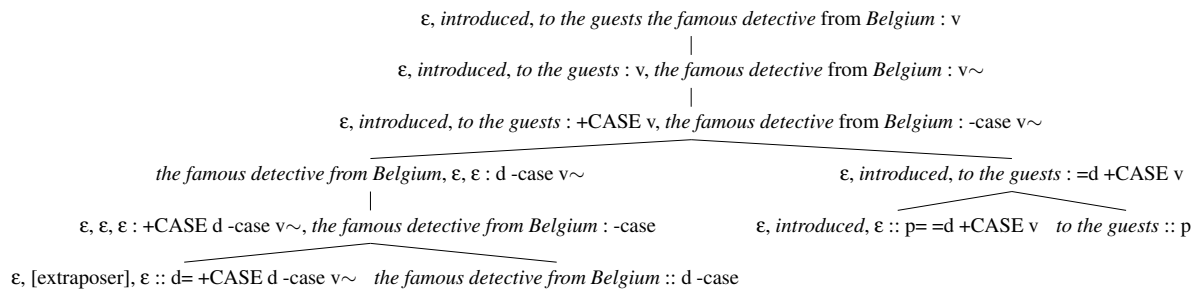
$$\begin{array}{c} \textit{the famous detective from Belgium}, \varepsilon, \varepsilon : d -case v\sim \\ | \\ \varepsilon, \varepsilon, \varepsilon : +CASE d -case v\sim, \textit{the famous detective from Belgium} : -case \\ \hline \varepsilon, [\textit{extraposer}], \varepsilon :: d= +CASE d -case v\sim \quad \textit{the famous detective from Belgium} :: d -case \end{array}$$

We now merge this DP with the V' we derived in the previous step, resulting in the new V' shown at the root of the following derivation tree.

<sup>22</sup>This strategy simulates the ‘pied-piping mechanism which is sometimes proposed within TG. It is suggested in the context of VP topicalisation by (Kobele, 2006, page 167), and is a welcome strategy because Kobele (2005) has shown that adding a genuine pied-piping mechanism whereby licensees are percolated from specifiers to heads results in type 0 MGs.



Finally, we apply Move2 to check the -case feature of the object DP, and then  $r\_move$  to check its  $v\sim$  licensee. The resulting VP is shown at the root of the following derivation tree.



The remaining steps of the derivation, which introduce the little  $v$  head, the subject and the tense and complementizer heads, are left as an exercise. Note that in the MGbank phrase structure trees, the trace of a rightward movement is marked by a  $\zeta$  in contrast to the  $\lambda$  and  $\Lambda$  symbols used for leftward phrasal movement and head movement traces respectively. The overtly pronounced terminals contained in the rightwardly moved item are enclosed in forward slashes, which are intended to indicate that these items have phonetic features, but not semantic ones, since the standard assumption is that rightward movement is a PF operation only. The derived Xbar tree for our example sentence is shown in fig 3.18.

At the start of this section we noted that it is only DPs which are phonetically ‘heavy’ which qualify for rightward movement; the concept of ‘heaviness’ is notoriously difficult to pin down in formally, but we can approximately define as ‘consists of a relatively large number of phonemes’. Grammars are not usually formulated so as to allow their rules to make direct reference to strings, and the EDMG presented here is no exception to this. However, MGbank does implement a very approximate version of the heaviness condition on rightward movement by banning such movement from applying to pronominals, which are all ‘light’. This successfully blocks cases such as 44b above, for instance. Looking ahead slightly to section 3.4, this is achieved in MGbank by placing associating a negative selectional requirement -PRO with the  $d=$  selector of the [extraposer] head, and by associating a PRO selectional property with the  $d$  selectee of all pronominals. The updated type for our null [extraposer] head and

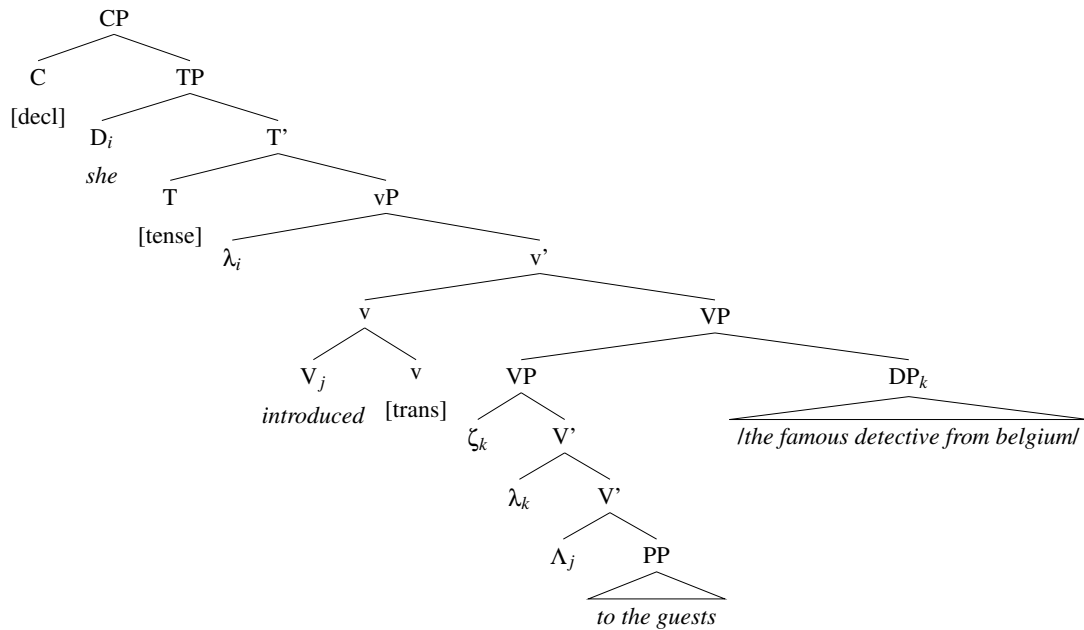


Figure 3.18: A (simplified) derived Xbar tree for the sentence *she introduced to the guests the famous detective from Belgium* which features heavy-DP shift.

for the pronoun *him* are given below.

[extraposer] :: d{-PRO} = +CASE -case v~

him :: d{PRO} -case

The -PRO requirement ensures that the [extraposer] head cannot be merged with any DP whose d selectee is associated with PRO; this therefore prevents it from merging with any pronominal and thus blocks rightward movement of this particular kind of 'light' DP.

### 3.3.4 Covert Movement/Move-F/Agree and !-type suicidal licensors

Prior to late Minimalism, the grammar was assumed to include so-called *covert movement* operations. Covert movement is movement which is not visible in the surface string because it takes place after SS (in GB) or Spell-Out (in early Minimalism). Two classic examples of phenomena which covert movement has often been used to describe are *wh-in-situ* constructions and quantifier raising. In English, where a sentence contains a single *wh* expression, that item appears in a preposed position either at the beginning of the clause that initially contains it (for embedded questions) or at the front of the entire sentence (for main clause questions). However, in languages like Japanese

and Chinese, the *wh*-item remains in-situ, as illustrated by 51a and 51b below for the embedded and main clause cases respectively.

- (51) a. John-wa naze kubi-ni natta no (Japanese)  
 John-top why was fired Question-marker  
 ‘Why was John fired?’
- b. Wo xiang-zhidao la fn, Lisi mai-le sheme (Chinese)  
 I wonder Lisi bought-Asp what  
 ‘I wonder what Lisi bought’

As we have seen, Chomskyans argue that at a relevant level of abstraction, all languages can be shown to be variations on a common theme. One of the ways in which languages have often been hypothesised to be invariant within this framework is that different languages will have the same LF structure for a given construction type. This means, for instance, that if a *wh*-word moves in a given context in one language, it must move in that same context in all languages. Where languages can differ parametrically is in whether or not that movement takes place overtly or covertly. The *wh*-words in examples 51a and 51b are therefore assumed to have undergone covert movement to spec-CP of the matrix and embedded clauses respectively, whereas in English this movement is overt and therefore shows up in the surface string.

Where there is more than one *wh*-word in an English sentence, only one of the *wh*-expressions may move while the other remains in situ, as illustrated in 52 below.

- (52) Which customer bought which car?

Chomskyans have often argued that any *wh*-items which remain in situ in the surface string actually undergo covert *wh*-movement to multiple spec CP positions by LF. Again, the standard assumption is usually that such movement occurs in all languages, but that languages may vary parametrically according to whether one or all of the *wh* expressions moves overtly. In Polish, for example, all *wh* items move overtly.

- (53) Kto co robi? (Polish)  
 Who what does  
 ‘Who does what?’

Cases of multiple-*wh* constructions in English such as 52 are rare in the Penn Treebank and, as we saw in section 3.2.10, are not covered by the MGBank grammar at present owing to the effects of SMC. However, the MGBank grammar does include covert *wh*-movement operations to handle echo questions, such as *you said what!?*, in which the *wh*-word remains in situ but is emphasised.

Wh-words are generally argued to move to the left periphery of a clause to make them more prominent and so that they can establish scope over that clause. Other types of scope relations have also been argued by transformationalists to be represented in the syntax. For instance, covert movement has also standardly been used to account for the ambiguity present in sentences such as 54 below which contains both an existential and a universal quantifier.

(54) Some boy loves every girl?

This sentence means either that there is a single boy who loves every girl, or, for every girl there is a boy who loves that girl. In the surface syntactic structure, it is clear that *some boy* is structurally more prominent than *every girl* (because subjects c-command objects), and so the second interpretation is surprising. Chomskyans have traditionally accounted for this by assuming that quantified argument phrases undergo covert movement to adjoin to TP in order to establish their scope. The quantified phrase which receives the widest scope will be the one which moves to the outermost adjoined TP position. In other words, the two interpretations of 54 have the LF structures shown below.

$$[_{TP} [\text{some boy}]_i [_{TP} [\text{every girl}]_j [_{TP} t_i \text{ loves } t_j ] ] ]$$

$$(\exists y[\text{person}'y \wedge \forall x[\text{person}'x \rightarrow \text{loves}'yx]])$$

$$[_{TP} [\text{every girl}]_i [_{TP} [\text{some boy}]_j [_{TP} t_j \text{ loves } t_i ] ] ]$$

$$(\forall x[\text{person}'x \rightarrow \exists y[\text{person}'y \wedge \text{loves}'yx]])$$

While it would be perfectly possible to capture this sort of quantifier raising in the present formalism, it is not included in MGBank for the simple reason that quantifier scope ambiguities are not resolved in the PTB, hence it would have been impossible for the automatic treebank generator to determine scope relations in cases where there are both existential and universal quantifiers in the same clause. In practice, it is almost always the surface subject which takes wide scope, so not too much is lost in practice by this omission.

A further classic example of a construction that was often been treated as involving covert movement in the pre-Chomsky (2000) literature is so-called *expletive replacement*. Consider the following examples.

(55) a. Several problems remain.

- b. There remain several problems.

These examples appear to have (virtually) identical meanings. For this reason, *there* is usually considered to be a meaningless expletive pronoun when it appears in subject positions. However, as we have seen, TG assumes that if the same construction across two different languages have precisely the same meaning, then they should also have identical LF representations. The same must therefore be true of two different constructions which have precisely the same meaning within the same language.

Chomsky (1986b) argued that the semantic similarities between 55a and 55b above could be accounted for by assuming that the thematic DP moves at LF to replace the expletive, resulting in identical LF structures for these two examples. The other instances of covert movement we have looked at were covert A'-movements, i.e. movement to a non-argument position, but Chomsky was arguing here for an instance of covert A-movement. However, Chomsky (1991) subsequently pointed out that in fact the DP associate does not behave as though it were in the subject position for scope purposes, as the following examples illustrate.

- (56) a. Many problems did not arise (*many* has wider scope than *not*).  
 b. There did not arise many problems (*many* has narrower scope than *not*).

Chomsky (1995) therefore proposed a revised theory of covert movement according to which it actually takes place in overt syntax but involves movement of formal features only, without pied-piping of phonetic features. This was referred to as the Move-F approach to covert movement, and can be seen as an early attempt by Chomsky to eliminate the covert cycle from the grammar altogether (which he later argued to have done via the introduction of Phase Theory in Chomsky (2000, 2001b)). If only formal syntactic features move in expletive *there* constructions, then the scope differences between 56a and 56b are accounted for.

In the next section we will see that the Move-F approach was ultimately supplanted by Chomsky's 2000; 2001b Agree operation, which does not involve any covert movement at all. Instead, a probe searches down the phrase structure tree for a matching goal, with feature valuation and deletion taking place at a distance. Given that MGs take a derivational rather than phrase structural view of syntax, however, in which movement only applies to strings not to structures, it is difficult to see how covert movement/Move-F and Agree can be formally differentiated here, unless we wish to say that only the former has semantic effects.<sup>23</sup> Although the rules in this section will

---

<sup>23</sup>In Kobele's (2006) system, we would then associate the licenser triggers of 'covert movement' with

be referred to as covert movement rules, therefore, they could also be viewed as implementing a simple version of the long-distance operation Agree that uses early MP style feature checking, rather than the full blown feature valuation system of later MP.

The covert movement rules used here are taken from Stabler (1997). In order to allow chains to begin moving covertly, phonetic Merge rules are added to the grammar which split the dependent's head chain into its syntactic and phonetic parts. For example, merge4, merge5 and merge6 have the corresponding *phonetic merge* rules in fig 3.19, which fuse the selected item's string to the selecting item's head chain but keep the selected item's syntactic features in a separate moving chain. Similarly, Adjoin3 and Adjoin4 from section 3.3.1 have the corresponding phonetic Adjoin rules shown in fig 3.20;<sup>24</sup> in this case, of course, it is the selector whose string is fused in place and whose formal features begin to move covertly.

$$\frac{[\varepsilon, s_h, \varepsilon :: x = \gamma] \quad [t_l, t_h, t_r ; x \delta, \alpha_1, \dots, \alpha_k]}{[\varepsilon, s_h, t_l t_h t_r : \gamma, \varepsilon : \delta, \alpha_1, \dots, \alpha_k]} (p\_merge1)$$

$$\frac{[t_l, t_h, t_r ; x \delta, \alpha_1, \dots, \alpha_k] \quad [\varepsilon, s_h, \varepsilon :: = x \gamma]}{[t_l t_h t_r, s_h, \varepsilon : \gamma, \varepsilon : \delta, \alpha_1, \dots, \alpha_k]} (p\_merge2)$$

$$\frac{[t_l, t_h, t_r ; x \delta] \quad [s_l, s_h, s_r : = x \gamma, \alpha_1, \dots, \alpha_k]}{[t_l t_h t_r s_l, s_h, s_r : \gamma, \varepsilon : \delta, \alpha_1, \dots, \alpha_k]} (p\_merge3)$$

Figure 3.19: Three phonetic Merge rules enabling subsequent covert movement.

$$\frac{[t_l, t_h, t_r ; x \approx \delta] \quad [s_l, s_h, s_r ; x \gamma, \alpha_1, \dots, \alpha_k]}{[s_l, s_h, s_r t_l t_h t_r : x \gamma, \varepsilon : \delta, \alpha_1, \dots, \alpha_k]} (p\_adjoin1)$$

$$\frac{[s_l, s_h, s_r ; x \gamma, \alpha_1, \dots, \alpha_k] \quad [t_l, t_h, t_r ; \approx x \delta]}{[s_l, s_h, s_r t_l t_h t_r : x \gamma, \varepsilon : \delta, \alpha_1, \dots, \alpha_k]} (p\_adjoin2)$$

Figure 3.20: Phonetic Adjoin rules allowing the adjunct to subsequently undergo covert movement.

An item which is currently undergoing overt movement can also land phonetically but then continue to move covertly, and the rule describing this scenario is given in fig

some non-vacuous semantic value, and the licensors triggering 'Agree' with the identity function.

<sup>24</sup>p\_adjoin2 would be needed, for example, to capture an echo question in which the in situ wh item is an adjunct: You left the restaurant *when?*!



3.21. Note that there is no corresponding rightward movement rule here, because we are adopting the standard assumption that rightward movement only involves phonetic features.

$$\frac{[s_l, s_h, s_r : +f \gamma, \alpha_1, \dots, \alpha_{i-1}, t : -f \delta, \alpha_{i+1}, \dots, \alpha_k]}{[ts_l, s_h, s_r : \gamma, \alpha_1, \dots, \alpha_{i-1}, \varepsilon : \delta, \alpha_{i+1}, \dots, \alpha_k]} (p\_move)$$

Figure 3.21: The movement rule resulting in phonetic merge of the moving string and subsequent further covert movement of its formal features.

The formalism also allows for the possibility in which an item is phonetically merged and its head string undergoes head movement. Formulating the phonetic versions of the rules *merge\_hm2* and *merge\_hm4* from section 3.3.2 is left as an exercise.

The covert movement operations themselves simply use the standard *Move1* and *Move2* rules from section 3.2.6; but because the moving chains in question have empty strings, this movement has no impact on the string. Note that the parser will treat *all* moving chains which have a null string as being covert movers, regardless of whether they were formed via the phonetic Merge/Adjoin/Move rules; for instance, if a null proform, say, or a remnant VP with no overt constituents remaining inside it, began to move, both would automatically be treated as moving covertly by the system. This can be significant, because whether or not *Move1* and *Move2* permit the mover to be covert or overt is determined by the licensor feature involved: lowercase +f licenses covert movement and uppercase +F licenses overt movement, as in Stabler (1997). The licensee features used for both overt and covert movement, meanwhile, are the same, namely lowercase -f. This means that the system cannot know which option to pursue initially, and so it must pursue both, creating two state paths corresponding to the overt and covert movement options.<sup>25</sup>

In MGbank, covert movement rules are used to capture various linguistic phenomena involving long-distance dependencies which clearly do not involve overt movement. One example is polarity item licensing of items like *any*, *anything* and *much*. Such items are restricted to appearing in negative or interrogative contexts, as the following examples illustrate

(57) a. *Jack has denied something/\*anything/\*much.*

<sup>25</sup>As a result of this optionality in the system, Merge/Adjoin and Move can no longer be described as unitary functions, because for the same input they may produce different outputs. Instead, they must be viewed as *relations*.

- b. *Jack hasn't denied much/anything.*
- c. *Has Jack denied much/anything?*

An important property of polarity item licensing is that it appears to be dependent on there being a c-command relation between the licensing (interrogative or negative head) and the polarity item. As discussed in section 3.2.4, the c-command relation between an antecedent and its trace is viewed here (as in Hornstein (2009)) as purely derivative, arising out of the monotonic nature of structure building operations, including Move. The same is true of c-command relations between probes and goals: given the Specifier Island Constraint which is (with some exceptions - see sections 4.3.1.1 and 3.3.7.2) enforced here, together with the fact that adjuncts can only be merged onto a governor once that governor has already checked all of its licenser features (thereby revealing its selectee), a probe can only search into its complement in order to check its licenser features, not into a specifier or an adjunct. Furthermore, all licenser and selector features on the head must be checked before that head can itself be integrated into any other structure. Treating polarity item licensing as covert movement therefore allows us to derive its c-command requirement straightforwardly.

To illustrate how this works in practice, we will look at the derivation for the interrogative sentence *has Jack denied anything?* We will do this using the following lexicon.

Jack : d -case  
 anything : d -case -pol  
 denied : d= v  
 [trans] :: >v= =d lv  
 has :: lv= +CASE t  
 [int] :: >t= +pol! c

Notice that the polarity item licenser feature has a ! diacritic attached to it which we have not seen before. Licensers such as these will be referred to here as *deleting suicidal licensers* (they contrast with *non-deleting suicidal licensers* which are marked with a ? diacritic and are discussed in section 4.3.4 below). This type of licenser differs from a standard licenser just in those cases where there is no matching licensee feature present to check, in which case it will simply self-delete rather than causing the derivation to abort. Thus in all the move rules presented so far we assume that +f can be a !-type suicidal licenser, but in addition we have the rule in fig 3.22 which applies

exclusively to suicidal licensors (including the ?-type to be presented in section 4.3.4), in which +f!/? can be either an overt or covert ?- or !-type suicidal licensor, and both  $\gamma$  and  $\delta$  are feature sequence suffixes which may be empty.

$$\frac{[s: +f!/? \gamma, \alpha_1, \dots, \alpha_{i-1}, t: \delta, \alpha_{i+1}, \dots, \alpha_k]}{[s: \gamma, \alpha_1, \dots, \alpha_{i-1}, t: \delta, \alpha_{i+1}, \dots, \alpha_k]} (fcide)$$

Figure 3.22: The self deletion rule for suicidal licensors. There must be no active -f licensee in the derivation for this rule to apply. If there is an active -f licensee which does not satisfy the fine-grained selection properties (+NOM, -INF, +3SG etc) of the suicidal licensor, then the derivation will abort.

This unary rule does not involve any movement; instead, the ? simply self-deletes because no matching licensee is present.

In order to ‘match’, the licensee need only be of the correct general type, i.e. -case, -wh etc; if such a licensee is present, but it fails to meet the fine-grained selectional requirements of the licensor (see section 3.4), then the derivation will still abort. In other words, all of the st

Using a !-type licensor here enables us to use the same interrogative C head whether or not there is a polarity item in the sentence. For example, in the sentence *has Jack denied it?* the +pol! licensor will not locate a matching -pol feature to check against because neither the object nor the subject of this sentence is a polarity item. Instead of this causing the derivation to crash, the +pol! feature simply self-deletes, allowing the derivation to continue. This self-deletion operation is represented as a unary branching node in the derivation tree (but does not add any structure at all to the derived phrase structure tree). The derivation trees for both *has Jack denied anything* and *has Jack denied it* are given in figs 3.23 and 3.24.

The reader will observe that *anything* initially moves overtly to check the +CASE licensor feature of the verb. At this point the string component of the moving chain for *anything* fuses into the specifier slot of *denied* while its formal syntactic features remain separated inside the still moving chain. At this point this chain has transitioned into a covert mover which will later check the +pol! covert licensor housed on the interrogative head. Because this licensor is suicidal (as indicated by the ! diacritic), even when a polarity item like *anything* is absent from the derivation, it can simply delete itself thereby allowing the derivation to continue as occurs in fig 3.24. Furthermore, because non-interrogative complementizers lack pol! licensors, examples such as 58

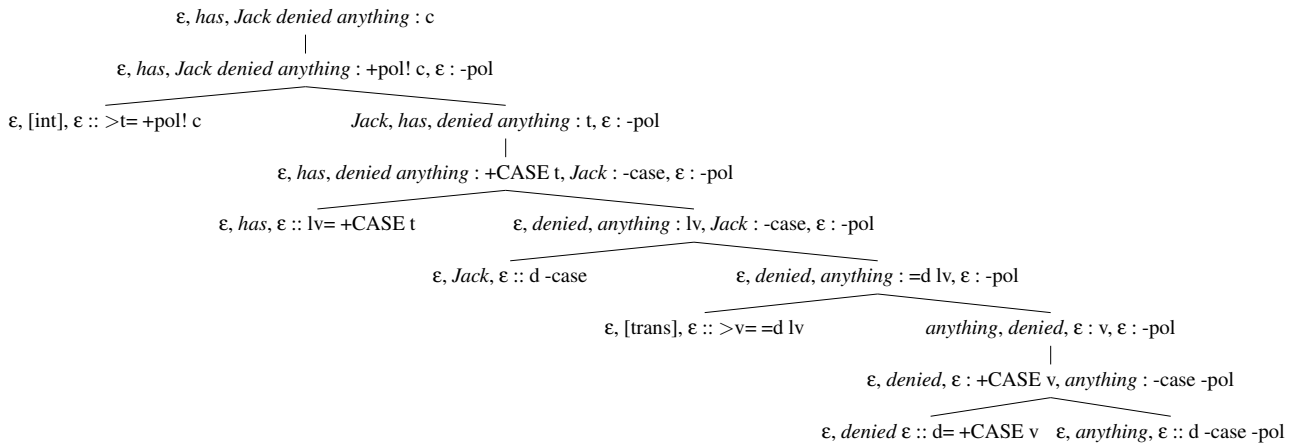


Figure 3.23: MG derived tree for the sentence *has Jack denied anything?* with the polarity item *anything* licensed by the interrogative *C* head via covert movement.

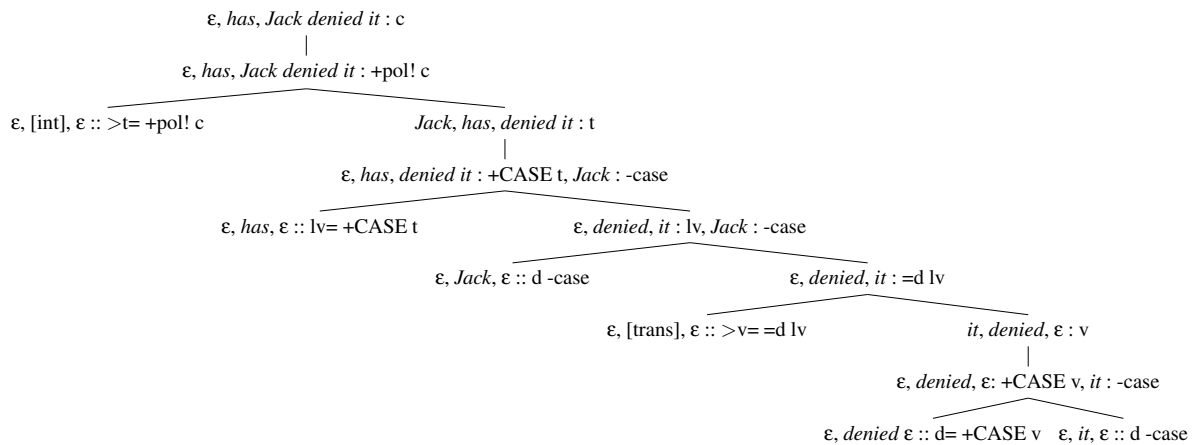


Figure 3.24: MG derivation tree for the sentence *has Jack denied it?* where there is no polarity item licensing.

below are correctly blocked.

(58) \*Jack has eaten anything.

There are also cases in which another argument may license the polarity item. Consider the following example.

(59) No one has eaten anything.

Again, the negative DP must c-command the polarity item in order to license it.

(60) a. No pictures have appeared anywhere.

b. \*Pictures of no one have appeared anywhere.

In MGbank, such examples are captured by placing a -negs licensee onto the negative determiner *no*. MGbank then uses a null [neg] head which appears along the clausal spine and carries both +negs and +pol! licenser features. In this way, [neg] heads can only be present to license a polarity item if there is also a negative DP in the derivation (as the +negs licenser is not suicidal and so must be checked and deleted by a matching -negs licensee), with the polarity item is thus indirectly licensed by the presence of said negative DP. Furthermore, owing to the Specifier and Adjunct Island Constraints, neither the -negs nor the -pol features can be embedded within a specifier or an adjunct, hence there will always be a c-command relation between the polarity item and the negative DP. Unfortunately, there is at present no way to prevent a configuration in which the polarity item is introduced into the derivation *after* the negative DP, meaning that the former will incorrectly c-command the latter. MGbank therefore currently overgenerates examples such as the following.

(61) \*Anyone has eaten nothing.

Another example of covert movement within the MGbank corpus is the case checking that is performed by prepositions. The derivation tree for the PP *to her* is given in fig 3.25 while its corresponding MG derived tree is given in fig 3.26. In all MGbank phrase structure trees, the landing site of a covertly moved item is marked by a  $\mu$ , and in the Xbar trees this item is also co-indexed with its overtly pronounced copy. Any terminals contained within an overtly pronounced copy of a covertly moved item are in MGbank enclosed in forward slashes (signifying that they are phonetically pronounced) and also written in capitals (signifying that they have semantic content, and distinguishing them from rightwardly moved items, which as we saw in section 3.3.3 are enclosed in slashes but written in lowercase).

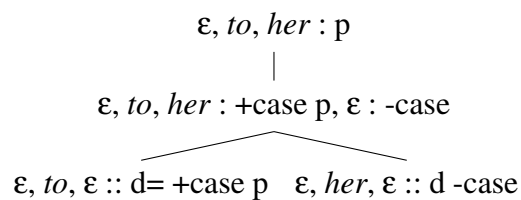


Figure 3.25: An MGbank-style derivation tree for the PP *to her*

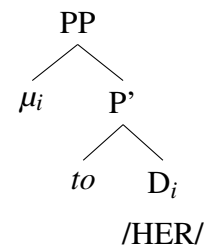


Figure 3.26: An MGbank-style Xbar tree for the PP *to her*

The reason why most adpositions in MGbank check case covertly is that English mostly has prepositions, not postpositions, and MGbank does not employ a shell structure for PPs the way it does for verb phrases, meaning that overt movement for case checking here would place the DP to the left of the P. There are, however, a limited number of postpositions in English (five miles *away*, five years *ago*, etc), and in MGbank these items have overt +CASE licensors instead of the usual overt +case ones.

One final point should be made before we move on to looking at how multiple Agree is handled in MGbank. In fig 3.23, *anything* moved overtly to check case, before moving covertly to check polarity. It would not have been possible for *anything* to move covertly to check -case because the parser would block this owing to the uppercase +CASE licensor on *denied*. However, the reverse situation can also arise, in which the lower licensor is a covert movement trigger while the higher one is a covert movement trigger. For example, consider the following sentence.

(62) *Who did you speak to?*

As we have just seen, prepositions check case covertly in MGbank. This means that *who* will first need to check the covert +case licensor on *to* before moving overtly to check the +WH licensor on the interrogative C which heads the sentence. However, once an item begins to move covertly, its string component will already have been fused in place, meaning that it cannot then convert back to moving overtly. For this reason, the parsers built for this thesis have an exception rule for covert movement licensors to the effect that a moving item may move overtly to check a covert licensor feature if they have more than one licensee feature remaining. This allows *who* to first move overtly to check the +case feature of *to*, before moving overtly a second time to its left peripheral position to check +WH. This exception holds only for covert licensors; an overt licensor can never be checked via covert movement.

As was noted above, licensee features are not divided into overt and covert versions (such as -CASE and -case) as licensors are. This design decision was made in order to avoid the situation where one would need to have two versions of many lexical entries, for example a version of *him* which moves overtly to check -case (when it is the object of a verb) and a version which moves covertly to check -case (when it is the object of a preposition). As we saw, as a consequence of this there is some optionality in the system because the system must initially explore both the overt and covert movement options. An alternative approach worth exploring would be to introduce -case features onto DPs using null heads (which would correspond to case morphology in languages which have it), much as is done for A' features like -top and -foc in MGBank. This would not reduce the search space, of course, since the parser would still have to explore both covert and overt movement alternatives (i.e. it would have to try merging every bare DP with both the overt and covert case heads), but it would allow us to eliminate the optionality in the system, making Merge/Adjoin and Move once again unitary functions rather than relations.

### 3.3.5 Existentials and Multiple Agree in MGBank

As discussed in the previous section, the treatment of phenomena traditionally handled by covert movement has undergone several evolutions within Minimalism, from the inclusion of a covert movement cycle which takes place post Spell-Out, to movement of formal features alone (Move-F), to more recent Agree-based accounts which do not involve any movement at all. As was also noted, in the derivational MG framework, Move-F and Agree are arguably indistinguishable (the valuation mechanism of Agree aside). In this section, we look more closely at the motivation for treating existential constructions such as 63 below as involving some kind of A-movement or A-Agree operation. We also give details of the MGBank analysis of existentials, which is a somewhat experimental implementation of the Multiple Agree-based accounts of later MP.

(63) *There are several people in the garden.*

This construction has been the subject of intense theoretical debate in the GB and MP literature. Of particular interest is the relation that apparently holds between the expletive subject *there* and its DP associate *several people*. One manifestation of this relation is the so-called definiteness effect, which refers to the fact that the DP associate must be an indefinite.

(64) \**There are the people in the garden.*

Second, although *there* appears in the subject position, it is the post-verbal DP associate which determines number agreement on the finite verb.

(65) *There is/\*are a man in the garden.*

In this respect, the DP associate therefore behaves syntactically as though it were in the subject position, as in the corresponding non-existential construction.

(66) *A man is/\*are in the garden.*

As we saw in the previous section, prior to Chomsky (2000, 2001b), these agreement facts were explained in terms of covert A-movement of the associate to the subject position. For example, Chomsky (1986b) argued that the associate moves to replace the expletive in spec-TP (then spec-IP) at LF, while in Chomsky (1991) the associate is argued to move to adjoin to the expletive at LF. In the Move-F approach of early Minimalism (Chomsky, 1995, page 364), the expletive is an overt manifestation of a D feature which is merged into spec-TP directly to check an uninterpretable D feature on T. Meanwhile, the associate's formal features raise to adjoin to the T head in order to check the uninterpretable agreement features on T, with T also checking the associate's uninterpretable case feature. The N feature of the associate then raises further to adjoin to the expletive sat in spec-TP.<sup>26</sup>

In support of the hypothesis that the expletive and associate are related via A-movement is the fact that this relation must be local. For example, just as overt A-movement is prevented from crossing a finite clause boundary in English (67c), so too is the expletive associate relation (68b).

- (67) a. [A man]<sub>i</sub> seems to t<sub>i</sub> be in the garden.  
 b. It seems (that) a man is in the garden.  
 c. \*[A man]<sub>i</sub> seems [<sub>CP<sub>fin</sub></sub> (that) t<sub>i</sub> is in the garden].

- (68) a. There seems to be a man in the garden.

<sup>26</sup>This latter operation is somewhat bizarre, as it involves movement from a position of adjunction to one head H to adjoin to the head of a (specifier) dependent of H. (Chomsky, 1995, page 364) claims that this is possible because 'in the relevant formal respects' the spec- $\alpha$  relation is the same as the head-complement one. While this is certainly true from the perspective of phrase structure (the specifier c-commands  $\alpha$ , which here is T', just as a head c-commands its complement), the two configurations are very different from a derivational perspective, and the movement Chomsky proposes is in fact impossible in the current formalism.



- b. \*There seems [ $CP_{fin}$  (that) a man is in the garden].

Furthermore, in languages and dialects which allow transitive expletive constructions, such as Belfast English (Henry and Cottell, 2007), the expletive must be associated with the closest DP to it in the structure. In Belfast transitive expletives, the associate DP must also be a quantified DP. Thus while 69 is allowed, 70 is not, indicating that the higher DP prevents *there* from associating with the lower one.

(69) There should lots of students get distinctions.

(70) \*There should the students get lots of distinctions.

The same type of intervention effect is found with standard overt A-movement. Thus 71b below can only mean that it is Pete who is expected to get a distinction, not Jack (excluding a purposive reading of the infinitival clause).

(71) a. Jack<sub>*i*</sub> expects to t<sub>*i*</sub> get a distinction.

b. Jack<sub>*i*</sub> expects Pete<sub>*j*</sub> to t<sub>*j*/*\*i*</sub> get a distinction.

However, we saw in the previous section that the hypothesis that the associate undergoes covert A-movement to the expletive is undermined by the fact that the associate behaves semantically as though it remains in-situ for scope purposes. A further argument against the covert A-movement approach is made by Lasnik and Saito (1991) and Den Dikken (1995), who point out that the purported covert movement of the associate fails to establish the same binding opportunities as its overt counterpart.

(72) a. [Some linguists]<sub>*i*</sub> seem to [each other]<sub>*i*</sub> to have been t<sub>*i*</sub> given good job offers.

b. \*There seem to [each other]<sub>*i*</sub> to have been [some linguists]<sub>*i*</sub> given good job offers.

In 72a, raising of *some linguists* to the matrix subject position allows it to bind the reciprocal *each other*, whereas no such binding is possible in the ungrammatical 72b. This is unexpected if covert movement in 72b establishes precisely the same structure as 72a at LF. Chomsky (1995) attempts to resolve these issues by suggesting that only syntactic features move, with semantic features remaining in-situ, an approach which could very easily be adopted into an MG framework. However, Lasnik (1997) subsequently showed that Chomsky's feature-movement approach to deriving 72a is overly restrictive.

Considerations such as these led Chomsky (2001b) to abandon the covert A-movement approach to the relation between expletive *there* and its associate and to propose instead that this relation is only indirect, mediated by the T head which enters into Agree relations with both *there* and the DP associate. T Agrees at a distance in person and number with the DP associate, resulting in the deletion of both the case feature of the associate and the uninterpretable person and number features on T; *there*, meanwhile, is merged directly into spec-TP in order to delete the *epp* feature of T while also having its uninterpretable person feature (which is inherently valued as third person) deleted.

Direct merger of *there* into spec-TP is not possible in the current grammar because licensor features, such as the +CASE on T that is responsible for attracting the subject here, can only be checked and deleted by Move, not Merge. Furthermore, in Chomsky's system, *there* must act as a probe (owing to its uninterpretable person feature), this being possible because *there* c-commands T from its spec-TP position. As Radford (2004, page 305) notes, this is theoretically dubious given that in all other cases of Agree the probe is the head of the main structure, not a dependent, and indeed this is the only possible scenario permitted by the formalism presented here. Chomsky also needs to assume that T can continue to serve as an active goal for *there* despite the fact that it has already had its uninterpretable person and number features valued and deleted by the DP associate by the time *there* enters the derivation.

Bowers (2002) also points out that Chomsky's analysis of expletive *there* constructions incorrectly fails to block the derivation of transitive expletive constructions in standard English such as the following.

(73) \**There has someone eaten a bagel.*

There is nothing in Chomsky's analysis which prevents T from agreeing with both *a bagel* and *there* in the above, just as it does in 63. Bowers therefore argues that in fact *there* is not merged directly into spec-TP, but is first generated in spec-vP of an unaccusative verb. According to Bowers, only verb phrases which do not take an external thematic argument may generate an external expletive argument in this way, which explains why unaccusatives but not transitives may occur with *there* in standard English. T then serves as a probe and agrees simultaneously with both *there* and the associate DP. Radford (2004, page 303) also suggests that the definiteness effect may be accounted for if we assume that the indefinite associate DP lacks person properties, with *there* serving to value and delete the uninterpretable person feature of T.

Sabel (2000) meanwhile, takes up Chomsky's (1995) suggestion that expletive

*there* is an overt manifestation of a D feature (explaining why *there* shares its *th* onset with other definite determiners such as *the*, *this*, *those* etc), but argues that this D element originates inside a DP which takes the associate as its NP complement. In the case of 63 above this DP would have the following structure: [<sub>DP</sub> *there* [<sub>NP</sub> *several people*]]. The D component then undergoes overt A-movement to spec-TP, stranding the NP associate. The A-movement-like locality restrictions on the expletive-associate relation are thus here accounted for not via covert A-movement of the associate, but by overt A-movement of the expletive away from the associate. The definiteness effect is also explained, because the D feature which contains the definiteness property has broken away from the associate DP.

The MGBank analysis of expletives combines aspects of all of the aforementioned approaches. It assumes with Chomsky that *there* has a third person feature but no number, and that *there* satisfies an *epp* feature on T. It adopts Bowers' view that the expletive enters into probe-goal relations with both the expletive and the thematic DP associate. It also implements Radford's suggestion that the DP associate lacks a person feature and that this motivates the introduction of the inherently 3rd person expletive into the derivation (as otherwise T's person feature would go unchecked). Finally, it takes up Sabel's suggestion that the expletive begins its derivational life by forming a constituent with the associate (accounting for the definiteness effect), although in the current formalism the only way to allow the expletive to A-move away from the associate is to have the former be a dependent of the latter, rather than vice versa.

To implement this hybrid approach, we introduce the new licensees -pers, -num and -*epp*, which can be viewed as an articulation of the standard -case licensee into three component parts. -num will appear on the DP associate and will be checked using the covert licenser +num on T, while -pers and -*epp* will appear on expletive *there*, with T also having +pers and +EPP features to check and delete these features. Although +pers is in principle a covert licenser, the -*epp* behind the -pers on *there* must be checked overtly by +EPP and this will therefore force +num to also be checked overtly in this construction (these features are also used for locative inversion, where they are distributed differently, and in this case, +pers is indeed checked via covert movement).

We will derive the simple existential sentence, *there remain several problems*, using the following (simplified) lexicon.

*there* :: d{3} -loc{EDGE} -pers -*epp*

*remain* :: d= v

several :: n= d{INDEF}  
 problems :: n  
 [associator] :: d{+INDEF}= +case =d{+3} D -num  
 [intrans] :: >v= +LOC! lv  
 [pres] :: lv= +pers +num +EPP t  
 [decl] :: t= c

The features enclosed in curly braces and attached to the main structure building features are selectional *properties* and *requirements* and are introduced fully in section 3.4. Briefly, X indicates some property, while +X indicates that the selecting or licensing feature requires X to be present on the selectee or licensee feature. These selectional features are used in MGBank to enforce subject verb agreement, case agreements and various selectional restrictions. The derivation tree up to the TP layer for our example sentence is given in fig 3.27, where the CP layer is omitted simply to save space on the page.

The null associator head serves two purposes in this derivation. First, it selects the indefinite DP *several problems* as its complement and checks both the latter's d and -case feature (covertly), effectively pied-piping the d (as it has a d feature of its own to replace the one that was deleted) and converting the -case into the sequence -num -epp. The +INDEF on the d= selector of this null head ensures that only indefinite DPs can serve as associates in expletive *there* constructions, thereby enforcing the definiteness effect. Next, it selects a DP specifier with the selectional property 3. The only item with this property in MGBank is expletive *there*, since it is the only DP which has person but not number (in accordance with Chomsky's proposal). *There* is thus selected and because it has remaining licensee features it becomes a moving chain inside the associate DP.

Next the associate DP is selected as the complement of the unaccusative verb *remain* and begins to move covertly, its string component therefore being immediately fused into the complement position of *remain*. The resulting unaccusative VP is then selected for by the unaccusative little v head [intrans]. This [intrans] head has a deleting suicidal +LOC! licensor which will check and delete the -loc on *there*. The purpose of this feature is to ensure that (for standard English) only unaccusative verbs can take expletive *there* (or locative PP) subjects, since only [intrans] (and progressive *be*) in MGBank possess this licensor feature. Thus non-progressive transitive expletive constructions such as 74 below are correctly blocked for standard English.

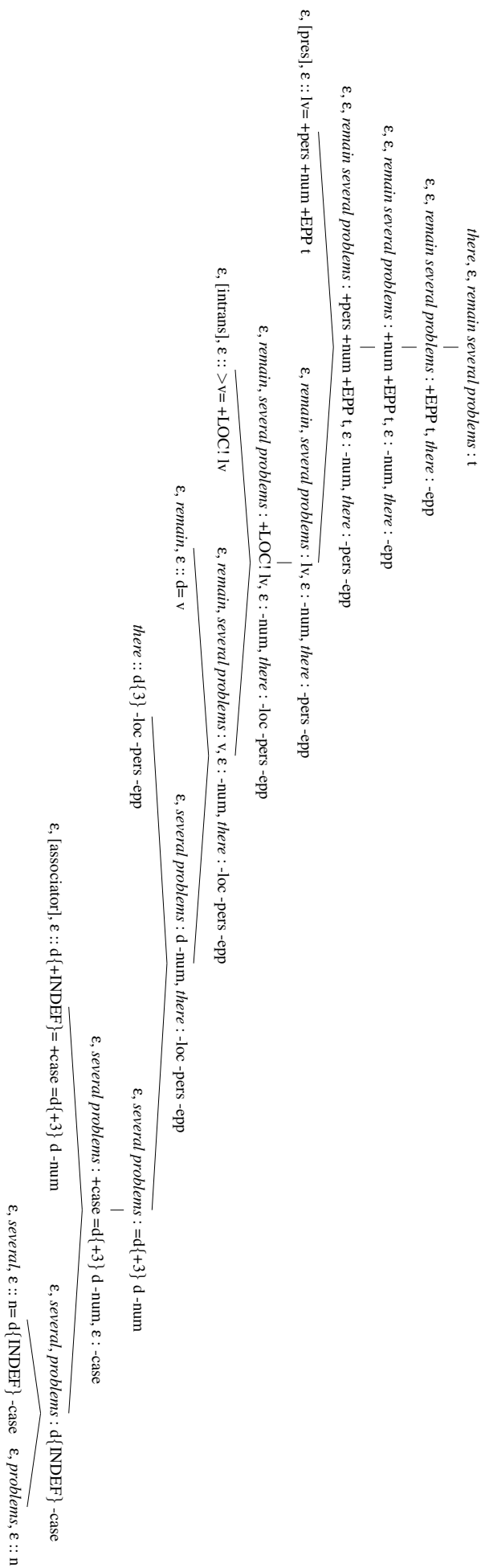


Figure 3.27: An MG derivation tree up to TP for the sentence *there remain several problems*.

(74) \*There should some students get distinctions (this year).

The progressive auxiliary *be* (in all its forms) has the following type, which also includes a +LOC! licenser.

*be* :: lv= +LOC! prog

This means that transitive expletives in the progressive aspect, such as 75 below, are correctly generated.

(75) *There are several students getting distinctions (this year).*

The reason the feature is called +LOC! is that it also licenses locative PP subjects which occur in very similar contexts to expletive *there* (see section 4.2.6). This approach of course simply stipulates the required distribution of expletive *there* in the lexicon. However, given that lacking (non-progressive) transitive expletive structures seems to be an idiosyncratic property of standard English (and not, for instance, of Belfast English), stipulating the result in the lexicon, the seat of arbitrary properties of individual languages, seems entirely appropriate.

The next step in the derivation merges the T head with the resulting vP. T then covertly checks the -num feature on the associate and overtly checks the -pers and -epp features on *there*. The reason why -pers is able to be checked overtly despite the licenser -pers being a covert licenser, is that, as discussed in the previous section, overt checking of covert licensers is permitted provided the moving chain in question currently has more than one licensee feature still to check. The final check of the -epp feature is the one which results in the actual movement of the string component of *there* to the leftmost position in the sentence string. Effectively, what we have here is a doubling approach to the relationship between expletive *there* and its DP associate, whereby these two items form a constituent early on in the derivation (at which point the fact that the associate DP must be indefinite is enforced), before *there* breaks away to satisfy the +EPP property on T.

One further point of note is the EDGE property attached to the -loc licensee. As will be discussed in more detail in section 4.3.1.1, an EDGE property on the active feature of a moving chain allows that chain to escape a specifier island. This is necessary wherever the expletive+associate compound is first generated in spec-vP, because *there* must be permitted to escape this specifier island on its way to the surface spec-TP position. As we shall see, the EDGE exception to the Specifier Island Constraint

is required for several other constructions which use a doubling analysis, including reflexive/reciprocal anaphor binding (see section 4.2.4) and the binding of floating quantifiers (see section 4.2.6).

Expletive *there* can also occur in other structural case positions in English, including being the object of an ECM verb or of the prepositional complementizer *for*.

(76) *Jack expects there to be fairies in the garden.*

(77) *For there to be fairies in the garden would be a surprise.*

MGbank therefore also includes special versions of the categories for ECM verbs and complementizer *for* with the licenser +CASE replaced by the sequence +pers +num +EPP, as was done for T. At this point it is perhaps worth asking whether it would not be better to simply eliminate the case features altogether, replacing -case with the sequence -pers -num -epp on non-associate DPs. We could then eliminate all the licensing heads with +CASE, leaving only those with the sequence +pers +num +EPP. In this new system, it is the feature -epp which triggers overt movement to nominative or accusative case positions, precisely as in current Agree-based Minimalism. However, case continues to play a role in current MP in that it *activates* a DP as a goal for feature checking purposes (owing to the fact that case is uninterpretable at the interface and hence must be checked and deleted). As we shall see in section 3.4.1, associated with -case licensees in MGbank are specific case properties (NOM, ACC and/or GEN) and agreement properties (1SG, 2SG, 3SG, 1PL, 2PL and/or 3PL). These agreement properties are then used by the licensing head (T or V or P) in order to ensure that the DP it attracts has the appropriate morphosyntactic agreement/case properties. In a sense then, -case is indeed used here as an activation feature, with the case and agreement properties being the features which are actually ‘checked’.

In the MG formalism, however, all licensee features serve to activate a goal, including the features -pers and -num which Chomsky would regard as interpretable as they appear on DPs (in contrast to the uninterpretable licensors +pers and +num appearing on verbal/prepositional heads). As has already been noted, here we are taking the view that all syntactic features are uninterpretable in the sense that they are not semantic features. However, some syntactic features are associated with non-vacuous semantic features (following Kobele (2006)), making them ‘interpretable’ in that sense.

One feature which seems not to be associated with any meaningful semantic value is the somewhat infamous -epp feature. But what is this -epp feature? EPP is the acronym for Extended Projection Principle, which was one of the Principles of GB.

Like all principles, it was essentially just a stipulation, in this case enforcing the constraint that all clauses require a subject. In later Minimalism, this Principle was reinterpreted as an abstract morphosyntactic feature in need of checking, although Epstein and Seely (2006) argue at length that it remains just as stipulative as it ever was, since *-epp* does not seem to correspond to anything in either the semantics or the morphosyntax. Note, however, that in the MGBank grammar, when *-case* is split into three features, the case specification property (NOM, ACC and/or GEN) ends up on the *-epp* feature (allowing e.g. finite T to attract only a nominative subject, and ECM verbs and *for* to attract only an accusative object). In other words, assuming that we were to adopt a system in which *-case* is always replaced by *-pers* *-num* and *-epp*, the categories for nominative *he* and accusative *him* would be as follows.

he :: D *-pers*{3} *-num*{SG} *-epp*{NOM}  
 him :: D *-pers*{3} *-num*{SG} *-epp*{ACC}

But given that *-epp* is associated with the case specification, this suggests that we should reanalyse this feature as none other than the licensee *-case*.

he :: D *-pers*{3} *-num*{SG} *-case*{NOM}  
 him :: D *-pers*{3} *-num*{SG} *-case*{ACC}

This brings us back full circle to Case Theory and Vergnaud's original idea that DPs undergo A-movement to delete case. Since case has observable morphosyntactic reflexes in language, I regard case-driven movement as a theoretically sounder idea than movement to check some mysterious *epp* feature. At present, however, the MGBank grammar continues to use *-case* alongside the three features *-pers* *-num* and *-epp* in the manner described here.

### 3.3.6 Raising and Obligatory Control

It is well known that the superficial similarity between raising and control examples such as 78 and 79 below belies important semantic and structural distinctions.

(78) *Jack seems to help Mary.*

(79) *Jack hopes to help Mary.*

For instance, it is clear that while in 79 *Jack* is the AGENT of both matrix *hope*



and embedded *help*, in 78 he is only the AGENT of *help*, with *seem* taking no external thematic argument at all. Thus only *seem* is permitted to take the expletive (i.e. meaningless) pronoun *it* as its subject.

(80) *It seems that Jack is helping Mary.*

(81) \**It hopes that Jack is helping her.*

Furthermore, while 78 entails 82, with the lower clause passivised, 79 does not entail the similarly passivised 83.

(82) *Mary seems to be helped (by Jack).*

(83) *Mary hopes to be helped (by Jack).*

One approach to distinguishing raising and control is to assign these two constructions the following (simplified) structures.

(84) [<sub>TP</sub> Jack<sub>i</sub> [<sub>vP</sub> seems [to [<sub>vP</sub> t<sub>i</sub> help Mary]]]

(85) [<sub>TP</sub> Jack<sub>i</sub> [<sub>vP</sub> t<sub>i</sub> hopes [to [<sub>vP</sub> PRO/t<sub>i</sub> help Mary]]]

In 84, *Jack* is initially generated in spec-vP of the embedded clause before moving to spec-TP of the matrix clause. As we saw in section 3.2.7, spec-vP is the position where AGENT DPs are generated and assigned their theta role, whereas spec-TP is a non-thematic surface subject position. As such, the only theta role assigned to *Jack* in 84 is the AGENT role of *help*. By contrast, although in the surface structure of 85 *Jack* again sits in the non-thematic spec-TP position of the matrix clause, he is co-indexed both with a movement trace in spec-vP of *hope* and with a null argument which is for now agnostically labelled as PRO/t in spec-vP of the *help* clause; *Jack* is thus interpreted as being the AGENT of both the matrix and embedded clauses in 85 (Note that on some versions of MP, DPs also move through/to spec-TP of infinitival *to*. In the MGBank grammar, *to* does not attract any DPs to its spec, however, for reasons which are discussed in section 4.7.3).

The reason for the ambiguous marking of the null subject in 85 is that the MP community is divided over whether this obligatorily controlled item should be treated as an A-movement trace, or whether it should be analysed as a null pronominal (referred to affectionately in the literature as *big PRO* to distinguish it from the so-called *little pro* subject of finite clauses in pro-drop languages such as Spanish). In the latter case, the co-indexation between obligatorily controlled (OC) PRO and its antecedent would

be established through control, which is often regarded as a type of binding, rather than via movement. In a Minimalist setting, Chomsky has claimed that binding ‘is at the outer edge of the C-I interface’ (Chomsky 2005b: 8), which seems to imply that it is not part of the syntax proper in the same way that standard A-movement is considered to be for example. There is some compelling evidence, however, to think that A-movement, binding and obligatory control should all be treated using the same basic mechanism. For example, both reflexive anaphors and obligatorily controlled null subjects adhere to many of the same locality constraints as the traces of A-movement, as the following examples illustrate.

- (86) a. Jack believes himself to be intelligent.  
 b. \*Jack’s mother believes himself to be intelligent.  
 c. \*Jack believes himself is intelligent.  
 d. \*Jack believes Mary to love himself.
- (87) a. Jack<sub>i</sub> hopes to t<sub>i</sub> be intelligent.  
 b. \*Jack<sub>i</sub>’s mother hopes to t<sub>i</sub> be intelligent.  
 c. \*Jack<sub>i</sub> hopes t<sub>i</sub> is intelligent.  
 d. \*Jack<sub>i</sub> persuaded Mary to t<sub>i</sub> help.
- (88) a. Jack<sub>i</sub> was believed to t<sub>i</sub> be tall.  
 b. \*Jack<sub>i</sub>’s mother was believed t<sub>i</sub> to be tall.  
 c. \*Jack<sub>i</sub> was believed t<sub>i</sub> is tall.  
 d. \*Jack<sub>i</sub> seems that it is believed to t<sub>i</sub> be tall.

In the (a) examples above the bound item (i.e. the reflexive, controlled null subject, or A-movement trace) has a local c-commanding antecedent and the sentence is grammatical. The (b) examples are ungrammatical (under the indicated co-indexation) owing to the fact that *Jack* is now embedded inside a larger DP and hence fails to c-command the anaphoric item. Example 88c illustrates the phenomenon of the so-called Nominative Island Condition, according to which A-movement is prohibited out of a tensed clause; examples 86c and 87c show that this island effect also shows up for binding and obligatory control. Finally, in each of the (d) examples, another DP intervenes between the antecedent and its intended anaphor, resulting in ungrammaticality (See Hornstein (2001) and Boeckx et al. (2010) for more extensive (semantic) arguments that control and the binding of reflexives/reciprocal anaphors should be treated as involving A-movement).

It was observations such as these which led to the classification of A-traces, PRO and reflexive/reciprocal anaphors in Chomsky (1981) as +ANA and therefore all subject to Principle A of the binding theory. As Boeckx et al. (2010) discuss, the primary reason for distinguishing PRO from traces of A-movement in GB was that all thematic relations were by definition expressed only at D(eep) S(tructure), which was generated using context free Xbar rules before any movement had applied. GB's approach to the unification of reflexives/reciprocals, PRO and A-movement traces was therefore to regard them all as types of anaphors, rather than treating them all as types of A-movement traces.

In Minimalism, on the other hand, DS has been eliminated and Merge (which is roughly equivalent to the old Xbar rules) and Move (equivalent to GB's move- $\alpha$ ) now freely intersperse, which opens up the possibility of assigning theta roles via movement. Moreover, in mainstream Minimalism (though not in the derivation-tree centred MG formalism), traces are now regarded as copies of their overt antecedents, rather than as special grammatical formatives with distinct features. Finally, Chomsky's conception of Move as a species of Merge seems to argue against treating External and Internal Merge differently by stipulating that thematic relations can only be established via the former.

Partly for these reasons, a substantial minority of Minimalist researchers have argued that binding and/or control should be analysed as types of A-movement (see, e.g., Hornstein (2001), Lidz and Idsardi (1998), Kayne (2002), Zwart (2002) and Boeckx et al. (2010)). In the case of control, this approach is sometimes referred to as the *Movement Theory of Control* (MTC). As we have already seen, in the MG presented in this dissertation, the only means by which to establish long-distance dependencies in the syntax is via movement. Postulating a single mechanism for long-range dependencies is clearly preferable on both methodological and evolutionary grounds to postulating several mechanisms, and is therefore arguably the more Minimalist approach. MGbank therefore adopts the essentials of the MTC of Hornstein (2001) and Boeckx et al. (2010), along with a version of the movement approach to binding which is inspired by Hornstein (2001). In this section, we will focus on the additional rules which are required for control, but these same rules are also used in constructions involving reflexive/reciprocal binding, to which we turn in section 4.2.4.

One property of control that should be noted is that it can iterate without bound. Thus the following is clearly grammatical, even if it is a little contrived.

(89) Jack [<sub>vP</sub> t<sub>i</sub> wants to [<sub>vP</sub> t<sub>i</sub> try to [<sub>vP</sub> t<sub>i</sub> strive to [<sub>vP</sub> t<sub>i</sub> help ] ] ] ] ]

Under the MTC, *Jack* is first base generated in spec-vP of the most deeply embedded verb of which it is the AGENT (here *help*) where it picks up its first AGENT theta role. After this, *Jack* must pick up the three other AGENT theta roles by moving through the intermediate spec-vP positions of each clause before arriving at its final spec-TP position in the matrix clause. Given that the AGENT theta role of *help* is assigned by checking the *d* feature of *Jack* and the =*d* feature of the null light verb heading the vP (via External Merge), the most natural assumption is that *all* AGENT theta role assignments involving checking of *d* and =*d* features. We will therefore allow =*d* and *d* to check and delete one another via both External Merge *and* Internal Merge (Move). However, we clearly do not want to say that in 89 *Jack* has three *d* features, one for each of his AGENT theta roles, because we could go on stringing control clauses together without bound. We would then need an infinite number of lexical entries for *Jack* to accommodate all possible cases.

We will therefore follow Kobele (2006) in assuming that there is a version of the selectee *d* which may optionally persist after checking, rather than being deleted.<sup>27</sup> Kobele marks this version of *d* as *d*\* (to distinguish it from *d* on expletives which cannot serve as controllers: *\*there/it wants to be a man in the garden*). Here we will mark the persistent version of *d* by using uppercase *D*. Thus the type of *Jack* will now be *D* -case.<sup>28</sup> When *Jack* is Externally Merged into the first spec-vP position in our above example, its *D* feature will optionally persist following checking. We then allow =*d* to be checked and deleted by the selectee *D* via Move. In other words, after its initial External Merge, *D* may optionally transition from being a selectee to being a control licensee (with =*d* now able to serve as a control licensor).

Note that (for reasons which are clearly rooted in semantics) only thematic DPs can be controllers, not expletive DPs and not other categories which are otherwise capable of being subjects, such as CPs and (arguably) locative PPs.

- (90) a. *Jack wants to help Mary*  
 b. *\*There/It wants to help Mary.*  
 c. *\*[CP That he succeeded] wants to help Mary.*  
 d. *\*[PP On the wall] wants to be hanging a picture of their ancestors.*

<sup>27</sup>Allowing persistent features has been shown to not increase the weak expressive power of MGs Stabler (2011).

<sup>28</sup>For simplicity, we assume throughout this chapter that proper nouns are bare determiner items. In the actual MGbank grammar, however, a proper noun is treated as a noun which is then selected for by a null determiner. It is therefore this null determiner which actually bears the feature sequence *D* -case.

(cf. the raising example: [<sub>PP</sub> *On the wall*] *seems to be hanging a picture of their ancestors.*)

We will therefore introduce a special set of Merge and Move rules which allow uppercase D features only to optionally persist. These rules are shown in figs 3.28 and 3.29, where  $\gamma$  and  $\zeta$  are feature sequence suffixes with  $|\gamma| \geq 0$  and  $|\zeta| \geq 0$ . All of the other Merge and Move rules presented so far can also apply as though D were a regular d feature (except that covert control movement is blocked by the MGBank parsers because English does not have backwards control - see section 4.2.5). In other words, D can optionally be checked and deleted like any other selectee, but it can also optionally persist, which is the scenario described by the special rules presented here.

$$\frac{[\varepsilon, s, \varepsilon :: d = \gamma] \quad [t_l, t_h, t_r; D \zeta, \alpha_1, \dots, \alpha_k]}{[\varepsilon, s, \varepsilon : \gamma, t_l t_h t_r : D \zeta, \alpha_1, \dots, \alpha_k]} (\text{merge\_ctrl1})$$

$$\frac{[t_l, t_h, t_r; D \zeta, \alpha_1, \dots, \alpha_k] \quad [\varepsilon, s, \varepsilon :: d = \gamma]}{[\varepsilon, s, \varepsilon : \gamma, t_l t_h t_r : D \zeta, \alpha_1, \dots, \alpha_k]} (\text{merge\_ctrl2})$$

$$\frac{[t_l, t_h, t_r; D \zeta] \quad [s_l, s_h, s_r :: d = \gamma, \alpha_1, \dots, \alpha_k]}{[s_l, s_h, s_r : \gamma, t_l t_h t_r : D \zeta, \alpha_1, \dots, \alpha_k]} (\text{merge\_ctrl2})$$

Figure 3.28: Merge rules with subsequent control movement

$$\frac{[s_l, s_h, s_r :: d = \gamma, \alpha_1, \dots, \alpha_{i-1}, t : D \zeta, \alpha_{i+1}, \dots, \alpha_k]}{[s_l, s_h, s_r : \gamma, \alpha_1, \dots, \alpha_{i-1}, t : D \zeta, \alpha_{i+1}, \dots, \alpha_k]} (\text{move\_ctrl})$$

Figure 3.29: Move rule in which D persists leading to further control movement

To see how these rules work in practice, the derivation tree for example 91 below, featuring two control movements, is given in fig 3.30. To enable the tree to fit on the page, the derivation omits the matrix clause complementizer.

(91) *He wants to try to help.*

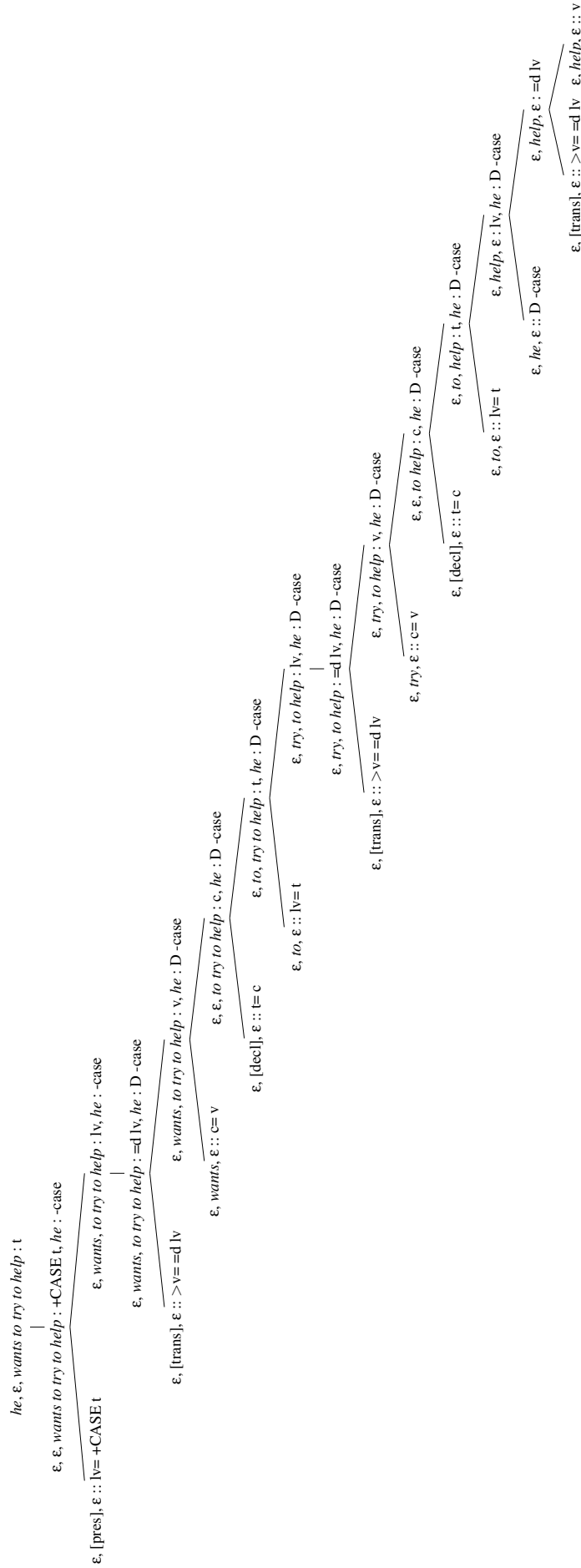


Figure 3.30: MG derivation tree for the sentence *He wants to try to help*. To keep the tree as small as possible, the matrix clause complementizer is omitted, showing the derivation only as far as the matrix TP.

### 3.3.7 Coordination and other related phenomena

In this section<sup>29</sup> we look at coordination within MGs, which has historically been considered problematic for TG. For example, Gazdar et al. (1985) claimed that “transformational grammar has never been able to capture a unitary notion of coordination, for reasons that were endemic to the framework.” Considered particularly troublesome are constructions which involve movement to a single position of two or more constituents which do not stand in a c-command relation with one another, as shown in schematic form in fig 3.31. This type of movement is referred to in the literature as *across-the-board* (ATB) movement.

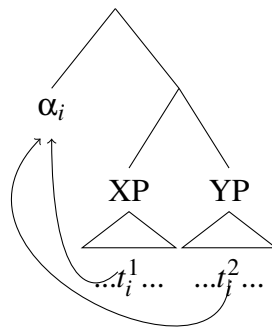


Figure 3.31: Across-the-board Movement Schema

Examples of constructions which have been argued to involve ATB movement are given below.

- (92) I know  $\text{who}_i$  [ $_{TP}$  Jack likes  $t_i$ ] and [ $_{TP}$  Mary hates  $t_i$ ].  
(ATB Phrasal Movement)
- (93)  $\text{Who}_i$  does $_j$  [ $_{TP}$  Jack  $t_j$  like  $t_i$ ] and [ $_{TP}$  Mary  $t_j$  hate  $t_i$ ]?  
(ATB Head and Phrasal Movement)
- (94) [ $_{TP}$  [ $_{TP}$  Jack likes  $t_i$ ] and [ $_{TP}$  Mary hates  $t_i$ ] [ $\text{Pete's sister}$ ] $_i$ ].  
(Right Node Raising)
- (95) He [ $_{VP}$  gave $_i$  [ $_{VP}$  Pete  $t_i$  a book] and [ $_{VP}$  Mary  $t_i$  a flower]].  
(Argument Cluster Coordination)

One approach to ATB movement has been to introduce a mechanism of *sideward movement* (Nunes 1995, 2001, 2004) into the grammar. This operation moves a constituent from one tree to another before those trees are merged together into a single

<sup>29</sup>The work presented in this section is based on Torr and Stabler (2016).

structure. For example, in fig 3.31,  $\alpha$  could first move sideways from  $t^2$  to  $t^1$  prior to the merger of XP with YP, before undergoing standard (upward) movement to its final surface position.

Two further constructions that have been argued to involve sideward movements, and hence the configuration in fig 3.31, are adjunct control and parasitic gap structures, as in 96 and 97 respectively.

(96) [<sub>TP</sub> He<sub>i</sub> [<sub>vP</sub> [<sub>vP</sub> t<sub>i</sub> filed the paper] [<sub>CP</sub> without [<sub>vP</sub> t<sub>i</sub> reading it]]]].

(Adjunct Control)

(97) [Which paper]<sub>i</sub> did [<sub>TP</sub> he<sub>j</sub> [<sub>vP</sub> [<sub>vP</sub> t<sub>j</sub> file t<sub>i</sub>] [<sub>CP</sub> without [<sub>vP</sub> t<sub>j</sub> reading t<sub>i</sub>]]]]?

(Parasitic Gap)

Example 96 features obligatory control by *he* of the null subject of the adjunct clause. Given our adopting of the MTC (discussed in the previous section), this means that *he* has successfully moved out of an adjunct in apparent violation of the Adjunct Island Constraint (AIC) (see section 4.3.1.2). In 97 there are two movements out of the adjunct clause, one control movement, just as in 96, and one *wh* movement. AIC is therefore apparently violated twice and yet this example is fine. However, under a sideward movement analysis, these movements can take place *before* the adjunct clause is actually merged as an adjunct, thereby circumventing AIC.

Stabler (2006) shows how a version sideward movement can be incorporated into MGs which can accommodate adjunct control examples such as 96. Unfortunately, this formulation of sideward movement is restricted to moving just a single element as an integral part of adjunction; as a result, it cannot accommodate example 97, which involves two elements moving out of the adjunct.

Kobele (2008) introduces an approach to leftward ATB phrasal movement for MGs which can accommodate these cases by ‘unifying’ any identical movers inside the dependent and main clause structures. However, Kobele does not extend his analysis to examples arguably involving ATB head movement (93 and 95) or Right Node Raising (RNR) (94). Moreover, as things stand, this system also appears to overgenerate 98 below, which features illicit ATB leftward phrasal movement from two different structural case positions.

(98) \*I know who<sub>i</sub> [<sub>TP</sub> Jack likes t<sub>i</sub>] and [<sub>TP</sub> t<sub>i</sub> hates Mary].

To handle such cases, we will adopt the essentials of Kobele’s system, but extend



it with mechanisms for rightward movement, excorporation, and case valuation.<sup>30</sup> Excorporation involves successive cyclic movement of a head past other heads and is argued to exist for example in Roberts (2010).

### 3.3.7.1 The Xbar Theoretic Approach to Basic Coordinate Structures

MGbank adopts the binary Xbar theoretic view of coordinate structures proposed most recently by Zhang (2010), in which the coordinator is the head, its complement is the rightmost conjunct and all leftward conjuncts are in specifier positions. For example, the derived Xbar tree structure for the phrase *Tom, Dick and Harry* is given in fig 3.32.

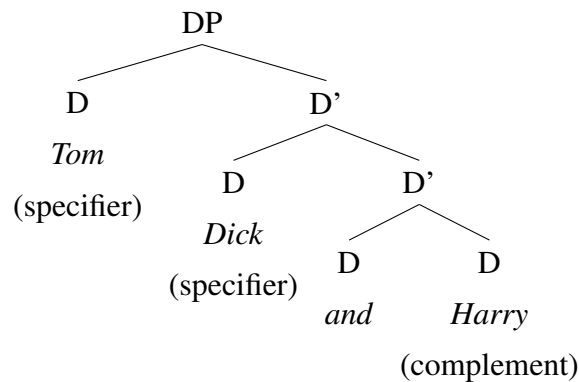


Figure 3.32: BPS tree with multiple specifiers for the coordinate phrase *Tom, Dick and Harry*.

Zhang assumes that coordinator heads inherit the part of speech category of their (leftmost) conjuncts, which here we will simply precompile into the lexicon by including separate coordinators whose  $x=$  selector and  $x$  selectee features share a common  $x$  category.

Coordination is a ‘recursive transitive closure over same types’ (Partee and Rooth, 1983), where ‘type’ in an MG refers to the  $::/$ : identifier plus the sequences of syntactic features on all of the chains of a given expression. The abstract type for all coordinator heads is given below.

$$s \overline{::} x = x \ x$$

The bar over the  $::$  is a diacritic identifying this head as a coordinator ( $\overline{::}$  is used for derived coordinator expressions). This is needed because the system must allow the

<sup>30</sup>Two important differences between Kobele’s framework and ours are: 1. We do not adopt a GPSG-style slash-feature mechanism; and 2. We do not handle control into complements via ATB movement (this is reserved for control into adjuncts)

=x specifier selector of coordinators to optionally persist after being checked so as to generate more than one specifier conjunct, as in fig 3.32 (in Torr and Stabler (2016) an overline on the selector was used to enable this type of persistence (i.e.  $=\bar{x}$ ), but since this can only occur for coordination, this is redundant once the  $\bar{\tau}$  feature is available for the parser to check). To allow for this, we must add the rule shown in fig 3.33 to the grammar.

$$\frac{[t_l, t_h, t_r ; x] \quad [s_l, s_h, s_r \bar{\tau} =x\gamma, \alpha_1, \dots, \alpha_k]}{[t_l t_h t_r s_l, s_h, s_r \bar{\tau} =x\gamma, \alpha_1, \dots, \alpha_k]} (\text{merge3})$$

Figure 3.33: The Merge rules for leftward conjuncts which allows the leftward selector to persist after checking in order to generate list coordination structures like *Tom, Dick and Harry*.

This type identifier is also used by the MGBank parsers in order to enforce the so-called Coordinate Structure Constraint (see section 4.3.2).

The present approach to coordination is somewhat similar to that of Combinatory Categorical Grammar (CCG) (Steedman (2000)), except that here it is not formally treated as involving adjunction (as it does not involve  $\approx$  type selectors). Full type uniformity is also not enforced by the selector features alone here, as it is in CCG, because MG selector features are never complex. For example, although we can specify that both conjuncts must be CPs using the feature sequence  $c = c$ , we are unable to define feature sequences equivalent to CCG's  $((S/NP) \backslash (S/NP)) / (S/NP)$ , in which both conjuncts are specified as being clauses containing object holes/traces. We can, however, enforce the constraint that a conjunct must have an identical set of moving chains as the main structure it is being merged into (see section 3.3.7.2); this, together with the Condition on Extraction Domains (CED) (see section 4.3.1) which bars extraction from specifiers (and clausal adjuncts, which is relevant for parasitic gaps and adjunct control), ensures that all conjuncts must be of like types in the sense of having the same types of movers inside them.

One problem for the analysis so far arises out of the fact that the MGBank grammar assumes that all DPs (with the exception of NOC PRO) bear -case licensees. This severely restricts the distribution of DPs, but is problematic for cases of DP coordination since every DP conjunct's -case feature will need to be checked. We cannot achieve this by adding covert +case features to the coordinator head because this would require the use of potentially infinite sequences of the form  $(=d +case)^+$ . To

solve this problem, we exploit a null prepositional dative head, independently used by MGParse to avoid SMC violations in *promise*-type subject control structures (see section 4.2). This head has the following type.

$$[\text{dat}] :: \text{d} = +\text{case p}$$

After this [dat] head checks the -case feature of its DP complement, the resulting PP is selected by the coordinator whose selector features are  $\text{p} =$  and  $=\text{p}$  but whose selectee feature is a D (this feature is uppercase to allow the coordinate DP to undergo control - see section 3.3.6). The type for *and* when coordinating DPs is given below.

$$\text{and} :: \text{p} = =\text{p D}$$

Notice that this implies that coordinators are able to inherit the part of speech category of their complement's complement (D), rather than that of the complement (P) itself. Since we do not formally treat coordination as adjunction, sacrificing this aspect of type-preservation becomes possible.<sup>31</sup> Interestingly, this move may not be entirely without empirical motivation: arguably, *Jack and me went home* is more natural than the prescriptively 'correct' *Jack and I went home*, as evidenced by the fact that *I and Jack went home* seems awkward, whereas *me and Jack went home* is informal but perfectly fine. This is explained if nominal conjuncts in English are in fact PPs with null dative case-checking P heads. In effect, we are still coordinating DPs, it is just that these DPs must have dative case and the [dat] head is the mechanism by which this is achieved in the MGBank grammar.

A further problem for the analysis of coordination presented so far is that there exist structures which do not appear to adhere to the like-types restriction on conjuncts, as exemplified by 99b below.

- (99) a. \*Jack [<sub>VP</sub> works] and [<sub>PP</sub> in the garden].  
 b. Jack is [<sub>VP</sub> working] and [<sub>PP</sub> in the garden].

As 99a indicates, coordination of a VP with a PP is generally not permitted, and yet in 99b it is allowed. It is in fact a general feature of the verb *be* that its complement can be a coordinate phrase with apparently unlike conjuncts. However, somewhat tellingly, only predicative categories can be coordinated following *be*. For instance, while *Jack is happy and in the garden* is fine, \**Jack is happily and in the garden* is ungrammatical

<sup>31</sup>We must, however, impose heavy restrictions in the lexicon to rule out many unwanted cases here.

because *happily* is adverbial rather than predicative. In fact, only VPs, PPs, AdjPs and DPs can be coordinated in this way following *be*. One approach pursued in the literature (e.g. Jacobson 1987) is therefore to assume that the expressions entering into such coordinate structures are in fact of the same super predicate category. MGBank implements this using the null predicatizer heads shown below. Observe that as well as mapping its complement into a version of itself with a prd selectee, these [prd] heads generate the DP subject as its specifier.<sup>32</sup>

[prd] :: d= +case =d prd

[prd] :: >v= =d prd

[prd] :: p= =d prd

[prd] :: adj= =d prd

These are essentially unary functions which replace the selectee feature of their complement with a prd selectee.<sup>33</sup> We can then simply coordinate the resulting PrdPs in the usual manner.<sup>34</sup>

### 3.3.7.2 Across-the-board phrasal movement, adjunct control and parasitic gaps

Consider deriving just the embedded clause from example 92, given as 100 below.

(100)  $\text{who}_i$  [ $_{TP}$  Jack likes  $t_i$  and [ $_{TP}$  Mary hates  $t_i$ ].

In terms of the schema in fig 3.31, *who* corresponds to  $\alpha_i$  and the two TP conjuncts to XP and YP. Recall that our problem here is to derive the fact that the two traces have only one overt antecedent and yet neither c-commands the other. Adapting an approach in Kobele (2008), we can accomplish this as follows: first we construct each TP conjunct. This yields the following two expressions:

*Jack*,  $\epsilon$ , likes : t, who : -wh

*Mary*,  $\epsilon$ , hates : t, who : -wh

<sup>32</sup>At present, only coordinate complements of *be* are headed by a [prd] head in MGBank. However, in a future release, this approach will be extended to all complements of *be*. Not only would this remove the need for separate versions of *be* selecting for complements with d, adj, v, p and prd selectees, but it would also make it much easier to capture constructions such as *there are fairies in the garden*, in which *fairies* could be represented as remaining in situ in the spec-Prd position; effectively, the [prd] heads transform their complements into small clauses.

<sup>33</sup>A similar approach incorporating null adverbializing [adv] heads, e.g. with category [p= adv], accommodates coordination of unlike modifiers in MGBank: *Jack works happily and with great speed*.

<sup>34</sup>An alternative analysis treats unlike coordination as ATB head movement, e.g. of *be* out of multiple remnant coordinated verb phrases which it heads. This is straightforwardly implementable using the mechanism for ATB head movement introduced in section 4.2.

Next, we merge the right conjunct *Mary hates who* as the complement of the following conjunction.

and  $\bar{t} = t$

After feature deletion, this yields the following expression.

and, Mary hates  $\bar{t} = t$ , who : -wh

This is where things become interesting. Notice that when the conjunction head merged with its complement, the mover inside the complement was transferred into the resulting expression. If this were to also happen when we merged the specifier, the result would be an SMC violation as we would now have two elements in the same tree whose first feature was -wh. Moreover, transferring a mover out of a specifier also violates the Specifier Island Constraint (SpIC) (see section 4.3.1.1). To solve this, we will bleed both SMC and SpIC by allowing the system to simply delete any mover inside any dependent if that mover's features exactly match those of a mover already inside the governing structure.<sup>35</sup> Deleting the occurrence of *who* from the left conjunct and merging the latter into the main structure will then yield the following TP coordinate phrase, correctly containing only one occurrence of *who*.

Jack likes, and, Mary hates : t, who : -wh

We can now merge this expression with the following null interrogative head.

[int] ::  $t = +WH$  c

Which, after moving *who* to check the +WH licenser of this head, will yield the following CP expression with the correct surface word order.

who,  $\epsilon$ , Jack likes and Mary hates : c

The ATB Merge rules for specifiers are shown in fig 3.34.<sup>36</sup> These rules can also generate parasitic gap constructions in which the gap appears inside a specifier, as exemplified in 101 below.

<sup>35</sup>Alternatively, we can view operation as unifying two sets of movers.

<sup>36</sup>A variant of *mrg\_atb1* allows the  $=x$  feature to optionally persist and therefore to generate recursive coordinate structures with ATB movement, such as *who does Jack like, Mary hate and Pete despise?*

(101) which celebrity did [pictures of  $t_i$ ] disgrace  $t_i$ ?

The adjunct control and parasitic gap examples 96 and 97 can also be derived using the same ATB mechanism, only in this instance the deleted mover(s) is/are inside an adjunct; the relevant leftward Adjoin rules<sup>37</sup> are shown in fig 3.35 (the derivation for 97 can be found in Torr and Stabler (2016)). In these rules, the string ( $\alpha^s$ ) and syntactic ( $\alpha^f$ ) parts of the  $\alpha$  chains have been separated, and identity is enforced only on syntactic features. This is because the same language is generated whether or not we stipulate string identity,<sup>38</sup> but not doing so results in a standard MCFG rule and therefore a proof of MCFG-equivalence.<sup>39</sup> Note that by combining *mrg\_atb1* with the rightward movement mechanism introduced in section 2.3.2, the MGBank grammar is also able to generate RNR examples such as 94, as instances of rightward ATB movement.<sup>40</sup>

Note that where the specifier is a conjunct in the rules in fig 3.34, we enforce  $l = k$  whereas for non-conjunct cases and for the rules in fig 3.35,  $l \leq k$ . This ensures that the like-types constraint applies only to coordination and not to parasitic gaps or adjunct

---

<sup>37</sup>The formalism also allows ATB movement out of leftward adjuncts, but formulating these rules is left as an exercise.

<sup>38</sup>For practical purposes, however, we allow the parser to also enforce string identity, since otherwise many partial parses are generated in which a moving substring in the dependent is dropped which does not phonetically match some moving substring in the main structure, and such a strategy can clearly never result in the recognition of a sentence.

<sup>39</sup>If we view the syntactic part of the head chain plus the  $\alpha^f$ s as a single atomic category symbol, then all we are saying in effect here is that combining a category of type A with a category of type B results in a category of type C, which is no different from any other MCFG rule. Seki et al.'s (1991) lemma 2.2 shows that banning variables that become erased during a derivation has no effect on expressive power.

<sup>40</sup>That RNR involves across-the-board rightward movement has often been proposed in the mainstream TG literature (Ross (1967), Bresnan (1974), Hudson (1976), Maling (1972), Postal (1974, 1998) and Sabbagh (2003)), although this approach is not without its theoretical problems (for critical discussion, see, e.g., Gazdar (1981) and Abels (2004)). For example, *Jack must have played and Tom must have hummed similar songs* clearly does not mean the same thing as *Jack must have hummed similar songs and Tom must have hummed similar songs* (.). A popular alternative is to treat the gap in the second conjunct as ellipsis (Wexler and Culicover (1980), Levine (1985, 2001), Kayne (1994), Wilder (1997), Hartmann (2001)). However, from our perspective, the RNR analysis is more attractive because it gives us a way of explicitly capturing the dependency of the shared item on both of the heads inside the two conjuncts. Ellipsis in MGBank, on the other hand, is simply treated using null heads; these are assumed to be 'anaphoric' in the broad sense of that term, but their co-referents would need to be established post-syntax. This is also the case, for example, with (non-reflexive/reciprocal) pronouns in MGBank, which unlike reciprocal and reflexive anaphors, are argued in Hornstein (2001) to not be related to their antecedents (when present) by movement (although see Kayne (2002) for an alternative view). The view taken here is that the more dependencies which can be established in the syntax the better, partly because this means that there is less semantic work to do later on, but also because generating dependencies in the syntax means that they must adhere to syntactic constraints (such as (T)SMC); it also enables the statistical model to be conditioned on them. This is why control and binding are treated as movement in MGBank (see section 3.3.6), and also why a Kaynian-style promotion analysis of relative clauses was chosen (see section 4.2).

$$\frac{[t_l, t_h, t_r : x, \alpha_1^f, \dots, \alpha_l^f(\alpha_1^{s'}, \dots, \alpha_l^{s'})] [s_l, s_h, s_r : =x\gamma, \alpha_1^f, \dots, \alpha_k^f(\alpha_1^s, \dots, \alpha_k^s)]}{[t_l t_h t_r s_l, s_h, s_r : \gamma, \alpha_1^f, \dots, \alpha_k^f(\alpha_1^s, \dots, \alpha_k^s)]} \quad (\text{merge} - \text{atb1})$$

$$\frac{[t_l, t_h, t_r : x\delta, \alpha_1^f, \dots, \alpha_l^f(\alpha_1^{s'}, \dots, \alpha_l^{s'})] [s_l, s_h, s_r : =x\gamma, \alpha_1^f, \dots, \alpha_k^f(\alpha_1^s, \dots, \alpha_k^s)]}{[s_l, s_h, s_r : \gamma, t_l t_h t_r : \delta, \alpha_1^f, \dots, \alpha_k^f(\alpha_1^s, \dots, \alpha_k^s)]} \quad (\text{merge} - \text{atb2})$$

Figure 3.34: Specifier Merge rules with ATB mover unification

$$\frac{[s_l, s_h, s_r : x\gamma, \alpha_1^f, \dots, \alpha_k^f(\alpha_1^s, \dots, \alpha_k^s)] [t_l, t_h, t_r : \approx x, \alpha_1^f, \dots, \alpha_l^f(\alpha_1^{s'}, \dots, \alpha_l^{s'})]}{[s_l, s_h, s_r t_l t_h t_r : x\gamma, \alpha_1^f, \dots, \alpha_k^f(\alpha_1^s, \dots, \alpha_k^s)]} \quad (\text{ad join} - \text{atb1})$$

$$\frac{[s_l, s_h, s_r : x\gamma, \alpha_1^f, \dots, \alpha_k^f(\alpha_1^s, \dots, \alpha_k^s)] [t_l, t_h, t_r : \approx x\delta, \alpha_1^f, \dots, \alpha_l^f(\alpha_1^{s'}, \dots, \alpha_l^{s'})]}{[s_l, s_h, s_r : x\gamma, t_l t_h t_r : \delta, \alpha_1^f, \dots, \alpha_k^f(\alpha_1^s, \dots, \alpha_k^s)]} \quad (\text{ad join} - \text{atb2})$$

Figure 3.35: Right Adjoin rules with ATB mover unification

control constructions, where we only require the  $\alpha^f$ 's in the specifier or adjunct to be a (possibly empty) subset of those in the main structure. That parasitic gaps are not subject to precisely the same constraints as coordination structures is evident from the fact that it is possible to fill a parasitic gap, leaving just the trace in the main clause, as in *which paper did Jack file without reading its title*, whereas we cannot extract from one conjunct but not the others (*\*who does Jack like and Mary hate Pete*) (part 2 of the CSC),<sup>41</sup> we will therefore assume here that both parasitic gaps and ATB-coordinate structures involve the ATB mover-deletion mechanism, but differ in that only coordination is subject to a like-types constraint (meaning that the same number of movers must be present in both conjuncts) owing to its semantics.

Recall that example 98 featured illicit coordination of two conjuncts containing traces in different structural case positions. What is needed here is some way for the system to ensure that accusative and nominative wh movers are not unified for ATB movement. MGBank currently uses different nominative and accusative versions of argument wh words such as *who* and *what*. This is done, for instance, in order to enable the system to enforce *that*-trace and *do*-trace effects on wh movement (see sections 4.4.1.1 and 4.4.2). As a result, the -case and -wh licensees of these items are associated with either NOM or ACC selectional properties (see section 3.4.1) marking

<sup>41</sup>We must also ensure that for conjunct (but not other) specifiers, only `mrg_atb1` can apply, i.e.  $|\delta| = 0$ , and similarly for the comp merge rules in fig 3.10 if the complement is a conjunct. These restrictions capture the fact that while ATB extraction of the identical contents of conjuncts is possible, extraction of conjuncts themselves is not (*\*who<sub>i</sub> does mary like t<sub>i</sub> and t<sub>i</sub>, \*who<sub>i</sub> does Mary like Pete and t<sub>i</sub>*) (part 1 of CSC). We can enforce these restrictions in the parser via reference to the use of the bar diacritic on coordinator projections.

them as either nominative or accusative. To block 98, we can therefore simply allow the parser to make reference to these features when it performs its check to see if the moving chains are identical for the rules in fig 3.34.<sup>4243</sup>

### 3.3.7.3 On sideward movement

It is important to note that although the approach to ATB movement adopted here may seem like an implementation of the sideward movement operation sometimes proposed in the MP literature, it is actually much weaker (i.e. more constrained) operation than what is often assumed there. This is because in the version proposed here, movement of  $\alpha$  from within A to within B must happen as part of the same Merge or Adjunction rule which combines A with B (the same is true of Stabler's 2006 version, where sideward movement for adjunct control occurs as an integral part of the operation merging the adjunct clause with the main structure). This is often not the case with the sideward movement that is proposed within the MP literature. For example, Kiguchi (2002) and Hornstein and Kiguchi (2003) (cited in Boeckx et al. (2010)), propose a sideward movement analysis of the following so-called *PRO-gate* example, which they argue involves obligatory control.

(102) [[PRO<sub>i</sub> washing herself] delighted Mary<sub>i</sub>].

If the PRO inside the gerund subject clause in this example is OC PRO, as these authors suggest, then under the MTC it must be a trace of A-movement. But since the gerund clause subject is base generated in a position from which it c-commands the object (in the present framework, in spec-vP, with the object in comp-VP), this cannot simply be a case of standard upward A-movement.

Kiguchi (2002) and Hornstein and Kiguchi (2003) argue that such examples should be generated as follows. First, the gerund clause is generated with *Mary* as its subject (we leave aside the coreferencing of the reflexive here), at which point *Mary* moves out of the gerund clause to the object position of *delight*. The derivation then continues, with the remnant gerund clause being merged into the AGENT subject position (in our context, spec-vP) and then moving to the surface spec-TP subject position as

<sup>42</sup> The parser does not refer to any other selectional properties or requirements for these checks, however. The reason for this is that there are cases where we need two items with different derivation histories to be unified for ATB purposes. Sometimes the selectional properties and requirements may percolate up onto one of these ATB movers and betray a differential derivational past. Therefore, only structure building features and case properties are checked for the rules in figs 3.34 and 3.35

<sup>43</sup>Note that this strategy replaces the earlier case valuation mechanism that was proposed in Torr and Stabler (2016).



usual. The crucial point here is that *Mary* vacates the gerund subject clause *before* the latter is merged into the main structure. In fact, there can be an arbitrary amount of structure between the object position and the subject position in PRO-gate sentences, as illustrated below.

(103) [[PRO<sub>*i*</sub> washing herself] seems have been expected to delight Mary<sub>*i*</sub>].

This means that this stronger type of sideward movement cannot be integrated into a binary Merge operation, nor can it be encoded as a unary node in the derivation tree, as it clearly requires two argument expressions: the gerund subject containing the chain with the active licensee, and the main structure whose head chain contains the matching licensor. As we saw in section 3.2.5, the MCFG equivalence result of MGs rests on the fact that Move can be represented using unary branching nodes (because this means that the set of MG derivation trees is a regular set); this in turn, is a consequence of the SMC and the fact that one need only specify a single expression as an argument to Move. If this stronger version of sideward movement is included in the grammar, however, this property of MG derivation trees seems no longer to hold. This suggests that enriching the grammar with this stronger operation may extend the weak expressive power of the formalism beyond that of MCFGs.

For this reason, MGBank treats PRO in examples such as 103 as a null pronominal, i.e. as non-obligatorily controlled PRO (see section 4.7.2), rather than as a trace of A-movement. This is arguably reasonable given that 104, where PRO has no available antecedent, is fine.

(104) [PRO washing oneself] is a bore.

This indicates that in 103, the reading under which PRO is co-referent with *Mary* is simply very strongly preferred, rather than syntactically obligatory.

#### 3.3.7.4 Across-the-board head movement

We still need to derive 93 and 95, both of which by hypothesis involve ATB head movement: T-to-C in 93 and V-to-v in 95. Note that the head movement rules presented in section 3.3.2 are insufficient here because there the moving head fused immediately with the head it adjoined to, making head movement a highly local operation. In general, this appears descriptively correct since heads cannot usually skip other heads, as shown for instance by 105a below.

(105) a. Would<sub>*i*</sub> you t<sub>*i*</sub> have helped?

- b. \*Have<sub>i</sub> you would t<sub>i</sub> helped?

Travis' (1984) *Head Movement Constraint* (HMC) describes the highly local constraints on standard cases of head movement. The problem with this constraint for examples 93 and 95 is that there is a coordinator blocking the movement path of the rightmost head (just as *would* blocks the movement path of *have* in 105b). There are, however, arguably certain exceptions to the HMC, involving successive cyclic head movement, i.e. the passing of one head through the edge of another head without (permanently) adjoining to it. For example, Roberts (2010, page 207) argues that Romance clitics such as the pronoun *l'* in the Italian and French examples below have undergone excorporation.

- (106) a. *l'ha vista* (Italian) (Roberts, 2010, page 207)  
 her'he/she saw  
 'She/he saw her'.  
 b. *je l'ai vu* (French)  
 I him/her'have seen  
 'I saw him/her.'

These clitics behave like heads in being affixal and adjoining to other heads, but they are also capable of moving over much greater distances than typical heads and in this sense behave more like phrases; in the French example, for instance, the clitic has moved from the post verbal object position, past its governing verb, and adjoined to the auxiliary. This, Roberts argues, is achieved via excorporation, or successive cyclic head movement,<sup>44</sup> and here we will use this mechanism to capture cases of ATB head movement past an intervening coordinator head.

To see why we need this here, consider again example 95, repeated as 107 below with some of the proposed structure now shown.

- (107) [<sub>CP</sub> Who<sub>j</sub> [<sub>C</sub> does<sub>i</sub> [<sub>TP<sub>cd</sub></sub> [<sub>TP</sub> Jack t<sub>i</sub> like t<sub>j</sub>] [<sub>T'</sub> [<sub>T<sub>cd</sub></sub> t<sub>i</sub> and] [<sub>TP</sub> Mary t<sub>i</sub> hate t<sub>j</sub>]]]]]]?

The derivation for this sentence initially proceeds precisely as in our earlier derivation from the previous section (except that T is now expressed overtly as *does*). How-

<sup>44</sup>This contrasts with the more standard roll-up head movement whereby one head H<sub>1</sub> moves to adjoin to another head H<sub>2</sub>, after which the complex H<sub>1</sub>-H<sub>2</sub> head may undergo further head movement to the next head H<sub>3</sub> and so on. This sort of roll-up head movement has been used to account for universal affix orderings on verbs (see, e.g., Holmberg and Roberts (2013)) because each of the heads in the functional (TP) domain of the clause is by hypothesis located in a fixed position relative to other heads and it is these functional heads which end up as suffixes under a distributed morphology approach. Roll-up head movement is found in MGbank, although null heads are used in place of affixes. One example is in main clause question formation, where, for instance, the copula verb *is* undergoes v-to-T-to-C roll-up head movement, picking up the empty [pres] head along the way.

ever, when the conjunction head merges with its right TP conjunct, the T head (*does*) of that conjunct will become fused either with its dependents as before or (if there is a > head movement diacritic on the t= selector of *and*) with the Coord head, rendering it inaccessible for further head movement to C. The situation for the left conjunct is even worse because we simply do not have any rules for head movement out of specifiers (as this would violate SpIC). Our solution to this will be to extend the grammar with a mechanism for excorporation which allows the head of a complement to move successive cyclically through the governing head rather than incorporating with it, as shown schematically in fig 3.36.

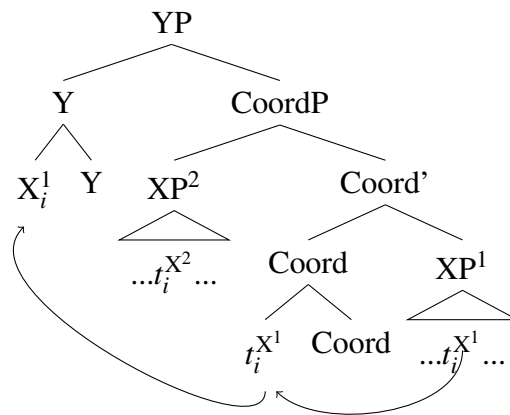


Figure 3.36: Excorporating ATB Head Movement

To implement excorporation, we add a new diacritic  $\hat{\ } to the selector which once again causes the complement's head to move. We then add conjunctions with the feature sequence:  $\hat{x} = \hat{x} x$  to the lexicon. This time, however, the raising head will become the new head of the selecting phrase, with the old head being fused onto the left end of the remnant complement string. This sets the stage for the new head to subsequently raise further, leading to successive cyclic head movement. The heads of any specifier conjuncts will simply be deleted,<sup>45</sup> just as their  $\alpha$  chains are deleted for ATB phrasal movement. The two rules are given in fig 3.38.<sup>46</sup> The first involves the complement case, hence the selector cannot yet contain any  $\alpha$  movers. The second rule shows the specifier case, and is rather like the specifier rule for ATB phrasal movement in that it involves dropping any  $\alpha$  chains in the selectee under feature identity with those in the selector. This time, however, the excorporation diacritic on the selector causes the head string of the selectee also to be dropped. Again only feature identity is$

<sup>45</sup>Co-indices on all head traces are deterministically recoverable from the derivation tree.

<sup>46</sup>Again, an additional rule is needed to allow  $=x$  to persist and derive, e.g. *who does Jack love, Mary hate and Pete admire?* Formulating this rule is left as an exercise.

required, hence the rule is MCFG-equivalent. The derivation for example 2 is given in fig 3.37 (only the leftmost conjunct's derivation is given in full).<sup>47</sup>

### 3.3.7.5 On the coordination of lexical $X^0$ heads

Xbar theory requires all complements and specifiers to be fully saturated, maximal XP projections. As pointed out in Borsley (2005), this poses a serious challenge to the Xbar theoretic view of coordinate structures, given that the coordination of unsaturated  $X^0$  lexical heads is apparently also possible, as the following example suggests.

(108) Hobbs [criticised and insulted] his boss.

In an attempt to rescue Xbar theory here, Kayne (1994) proposes that lexical coordination is only apparent, arguing that such examples feature ellipsis within the left XP conjunct. In other words, Kayne proposes that the analysis of 108 is *hobbscriticisedandinsultedhisboss* below.

(109) Hobbs [criticised ~~his boss~~ and insulted his boss].

However, as Borsley notes, there are other cases which do not appear amenable to this analysis. For example, 110a below clearly does not mean the same thing as 110b, a fact which also speaks against an RNR analysis of such constructions.<sup>4849</sup>

- (110) a. Hobbs whistled and hummed similar tunes.  
 b. Hobbs whistled similar tunes and hummed similar tunes.

(Boeckx and Jeong, 2004, page 84)

We will therefore take  $X^0$  coordination at face value, and in this section will propose a solution within the EDMG formalism that borrows from categorial grammar approaches and makes crucial use of the Earley-style dotted feature mechanism introduced in Kobele (2008). This mechanism enables features to remain visible to the system after being checked and 'deleted'.<sup>50</sup> This mechanism is illustrated for Move2 in fig 3.39.

<sup>47</sup>Note that the final step in this derivation is a unary rule fusing together the three string parts of a head chain iff it is the only chain in the expression, it covers all of the input string, and it has just one feature and that feature is a c (equivalent to reaching the S node).

<sup>48</sup>Lexical head coordination of this kind is in fact sometimes analysed as RNR in the PTB.

<sup>49</sup>As noted in fn 40, the problem is equally applicable to other cases of RNR involving coordination of phrases rather than heads. The categorial grammar style approach to lexical head coordination discussed in this section could be extended to cover all RNR cases in order to address this issue.

<sup>50</sup>Note that, unlike Kobele, we do not allow the system to make reference to already-deleted fea-

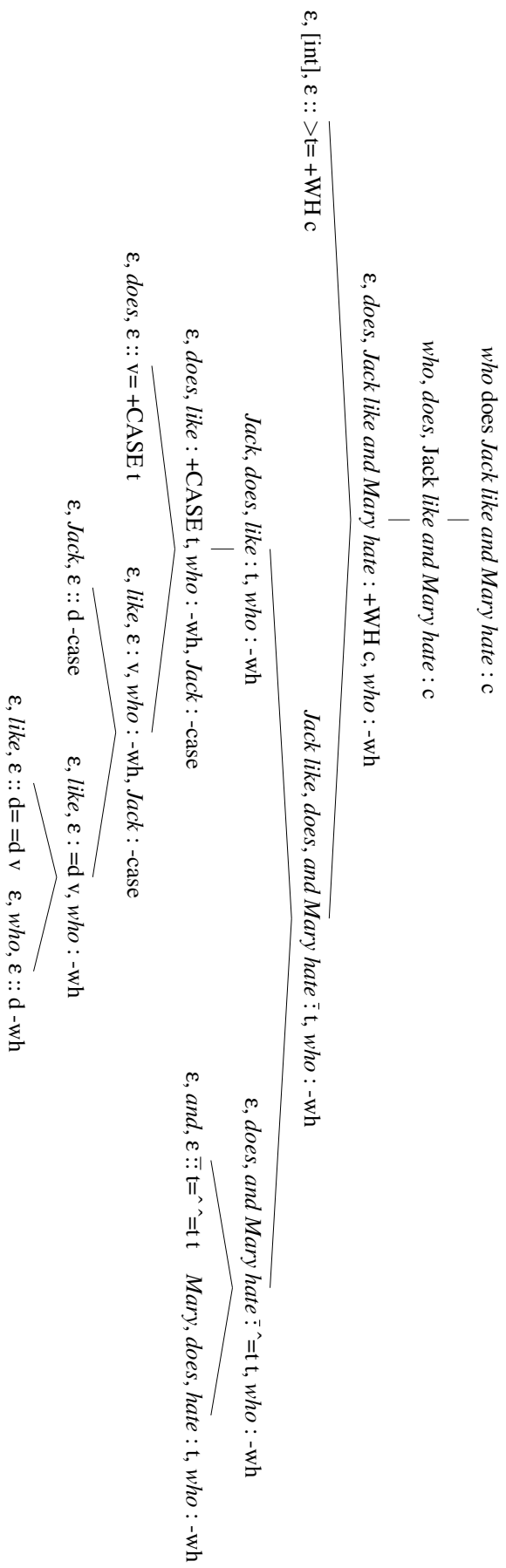


Figure 3.37: *who does Jack like and Mary hate*

$$\frac{[(\epsilon, s_h, \epsilon) \bar{\bar{\bar{\cdot}}} \hat{x} = \hat{x} \bar{\gamma}] \quad [(t_l, t_h, t_r) : x, \alpha_1, \dots, \alpha_k]}{[(\epsilon, t_h, s_h t_l t_r) \bar{\bar{\bar{\cdot}}} \hat{x} = \hat{x} \bar{\gamma}, \alpha_1, \dots, \alpha_k]} (mrg\_excorp)$$

$$\frac{[(t_l, t_h, t_r) : x, \alpha_1^f, \dots, \alpha_k^f (\alpha_1^{s'}, \dots, \alpha_k^{s'})] \quad [(s_l, s_h, s_r) \bar{\bar{\bar{\cdot}}} \hat{x} = \hat{x} \bar{\gamma}, \alpha_1^f, \dots, \alpha_k^f (\alpha_1^s, \dots, \alpha_k^s)]}{[(t_l t_r s_l, s_h, s_r) \bar{\bar{\bar{\cdot}}} \bar{\gamma}, \alpha_1^f, \dots, \alpha_k^f (\alpha_1^s, \dots, \alpha_k^s)]} (mrg\_hm\_atb)$$

Figure 3.38: Merge rules with excorporation and ATB head movement

$$\frac{[s : \beta \cdot +f \gamma, \alpha_1, \dots, \alpha_{i-1}, t : \zeta \cdot -f \delta, \alpha_{i+1}, \dots, \alpha_k]}{[s : \beta +f \cdot \gamma, \alpha_1, \dots, \alpha_{i-1}, t : \zeta -f \cdot \delta, \alpha_{i+1}, \dots, \alpha_k]} (move2 - dot)$$

Figure 3.39: Move2 with the dotted feature mechanism from Kobele (2008) added.

Within MGs, the Xbar theoretic requirement that all arguments must be maximal XP projections is encoded by the fact that all selectors and selectees must precede all selectee (and licensee) features (see section 3.2). That is, a given head must be fully saturated before itself being selected as a dependent, this being a crucial distinction between MGs and CCGs. In other words, taking  $\beta$  to be the sequences of selectors and licensors of a given chain, and  $\gamma$  to be its licensees, the only abstract head chain type which can be selected is the following, where the dot immediately precedes the selectee  $x$  feature:

$$[\beta \cdot x \gamma]$$

Assuming  $\beta$  to be non-empty, in the EDMG presented so far the above category could only have been derived via the application of Merge operations. Here we will introduce a new unary *type-saturation* mechanism whose argument is an unsaturated  $X^0$  head and whose output is a version of this head in which the dot has been moved to the left of the selectee feature. Such heads would not be semantically saturated at this point, of course, but this is fine provided that they can only be selected for by a special type of coordinator with a matching set of requirement (and licensee) features following its conjunct selector features; the matching requirement features on

---

tures as far as the identity checks for ATB movement are concerned. In other words, two items may have different derivational histories and still be unified for ATB movement provided they have identical unchecked feature building sequences (including identical fine-grained selectional properties and requirements associated with those features). This is needed in MGBank, for instance, in order to allow for the binding of reflexives inside adjunct clauses, as in *Jack helped Pete while also helping himself*, which is treated as ATB unification of a version of *Jack* which starts out inside a doubled constituent also containing the reflexive, with one which does not start out in this way - see section 4.2.4.1.

the coordinator projection can then subsequently satisfy the semantic requirements of all its  $X^0$  conjuncts in one fell swoop.

The rules for type-saturation and coordination of complement and (multiple) specifier  $X^0$  conjuncts are given in fig 3.40, where the asterisk serves the same function as the dot, except that it uniquely identifies type-saturated heads so that these are only ever selected for by a lexical head coordinator. Note that the rule h\_coord3 describes the scenario where the specifier conjunct selector feature persists after being checked, which allows us to generate examples with recursive head coordination such as 111 below.

(111) Hobbs criticised, dismayed, annoyed and insulted his boss.

$$\frac{[\epsilon, s, \epsilon :: \cdot \beta x \gamma]}{[\epsilon, s, \epsilon :: \beta * x \gamma]} \text{(type-saturation)}$$

$$\frac{[\epsilon, s, \epsilon :: \cdot x = \bar{x} \beta x \gamma] [\epsilon, t, \epsilon :: \beta * x \gamma]}{[\epsilon, s, t :: x = \cdot = \bar{x} \beta x \gamma]} \text{(h\_coord1)}$$

$$\frac{[\epsilon, t, \epsilon :: \beta * x \gamma] [s_l, s_h, s_r :: x = \cdot = \bar{x} \beta x \gamma]}{[t s_l, s_h, s_r :: x = \bar{x} \cdot \beta x \gamma]} \text{(h\_coord2)}$$

$$\frac{[\epsilon, t, \epsilon :: \beta * x \gamma] [s_l, s_h, s_r :: x = \cdot = \bar{x} \beta x \gamma]}{[t s_l, s_h, s_r :: x = \cdot = \bar{x} \beta x \gamma]} \text{(h\_coord3)}$$

Figure 3.40: Lexical head type-saturation and coordination rules

Clearly this approach is very close to the CCG analysis of lexical head coordination, in that it effectively allows unsaturated constituents to be selected for.

Notice that without the dotted feature mechanism, the subcategorization frame of a head would be lost following the type-saturation operation, as the  $\beta$  requirements would simply be deleted. However, we would then have no way to ensure that we were only coordinating heads of the same valency.

### 3.4 Incorporating I-selection and Agreement into MGs

Until very recently, all of the MGs proposed in the literature were devoid of any mechanism for capturing morphosyntactic agreement. As we saw in section 3.3.5, in the

later probe-goal MP framework, agreement is captured via the long-distance operation Agree, which involves valuation of uninterpretable features by interpretable ones. We also saw in that section how the long-distance mechanism of (multiple) Agree can be implemented using the covert movement rules from section 4.2.5 (which we could just as easily as labelled ‘Agree rules’ in view of the strongly derivational approach adopted here).

In this section<sup>51</sup> we look at how the MGbank grammar enforces specific morphosyntactic number, person and case agreement between agreeing items. This is implemented here as feature checking rather than feature valuation, in the sense that we will continue to assume that lexical items enter the derivation prespecified as nominative, accusative or genitive case, and as 1st person singular, or 3rd person plural etc (as in Chomsky (1995)), and that the system simply needs to ‘check’ that two items entering into a relation (via either Merge or overt/covert Move) are compatible with one another morphosyntactically.

The same system that is used to enforce morphosyntactic agreement here is also used to allow selectors and licensors to specify other fine-grained (semantic) selectional properties on the selectee and licensee features they check. Pesetsky (1996) refers to this type of fine-grained syntactic selection as *l*-selection (lexical selection), in contrast to the coarser grained syntactic *c*-selection (category selection), which MGs already encode via the =x/x= selector features, and semantic *s*-selection (semantic selection) (which could also be implemented using the system described here). For example, we may wish to specify that when the noun *promise* takes a PP complement, that PP should be headed by the preposition *to*, that the CP complement of the verb *wonder* should be interrogative, that the perfect auxiliary *have* should only take verb phrases headed by a past participle verb form, or that when the verb *want* takes a CP complement, that CP should be infinitival. The system to be presented also allows for certain apparent cases of long-distance selection, such as the fact that verbs like *demand* and *insist* require that the verb embedded inside their CP complements be in its bare form.

### 3.4.1 Case ‘Assignment’

Up to this point the nominative and accusative forms of personal pronouns in our grammar have not been distinguished. This means that as well as correctly generating *he*

---

<sup>51</sup>The work presented in this section is based on work presented in Torr (2018)



*helps him*, our grammar also overgenerates *him helps he*. One way to solve this would be to split +/-case features into +/-nom and +/-acc. However, many items of category D in English (e.g. *the, a, you, there, it*<sup>52</sup>) are syncretised for nominative vs. accusative case. This solution is therefore not very elegant as it would expand the lexicon with duplicate homophonic entries differing in just a single (semantically meaningless) feature. Furthermore, increasing the size of the set  $k$  of licensees could adversely impact parsing efficiency, given that the worst case theoretical time complexity of MG chart parsing is  $n^{2k+3}$  for MGs without head movement (Fowlie and Koller, 2017) and  $n^{2k+5}$  for MGs with head movement (Stanojević, 2019).

Instead, we will retain the single -case licensee feature and introduce NOM and ACC as subcategories, or *selectional properties*, of this feature. We will also subcategorise licensor features using *selectional requirements* of the form +X and -X, where X is some selectional property. Positive +X features require the presence of the specified property on the licensee feature being checked, while -X features require its absence. For example, consider the following updated lexical entries.

```

him :: d -case{ACC}
he :: d -case{NOM}
helps :: d= +CASE{+ACC} v{PRES.TRANS}
[pres] :: lv{+PRES}= +CASE{+NOM} t{FIN.PRES}
[trans] :: >v{+TRANS}= =d lv

```

The +ACC selectional requirement on the V head's +CASE licensor specifies that any matching -case licensee must bear an ACC selectional property, and similarly for +NOM on the T(ense) head. For SMC purposes, however, these two different subcategories of -case will still block one another, meaning that  $k$  remains unaffected. The reader should satisfy themselves that our grammar now correctly blocks the ungrammatical *him helps he*.

We can now also address the aforementioned syncretism issue without increasing the size of the grammar. To do this, we simply allow features to bear multiple selectional properties from the same paradigm. For example, representing the pronoun *it* as

---

<sup>52</sup>Minimalists assume many types of pronouns, including all personal pronouns, to be of category D(eterminer). Part of the reason for this is that many pronouns have the same phonological form as determiners, i.e. *that, this, these, those* and *some* can all be used prenominally or pronominally. This is also true of many personal pronouns; for example, *we, you, us* and (in certain non-standard dialects of English *them*) can all be used prenominally, as in *we republicans don't trust you democrats* (example taken from Radford (2004) page 46). MGbank follows this tradition and hence does not include a separate pronoun (PRP) category in contrast to the Penn Treebank.

follows will allow it to appear in either a nominative or an accusative case licensing position:

$it :: d \text{-case}\{\text{ACC.NOM}\}$

One complication is that if this -case licensee is checked by a +CASE licensor with, say, a negative -NOM requirement, we do not want the derivation to abort because *it* has an accusative version which just happens to be expressed by the same lexical item here. The formalism also allows for inclusive disjunctive features such as  $[+X|-Y|+Z]$ , which would be satisfied provided that either X and/or Z is present and/or Y is not present. One way around this problem would therefore be to use the disjunctive selector feature  $+[\text{NOM}|\text{GEN}]$ ,<sup>53</sup> instead of -ACC. However, the parser code currently contains another work around for this which is the following: for specific predefined paradigms, a negative selectional requirement -X on a licensor +f only blocks the derivation when checking a licensee -f associated with the selectional property X iff -f is not also associated with some Y, Y a member of the same *paradigm* as X, and +f is not associated with a -Y feature. In other words, a +CASE{-NOM} licensor would be able to check a -case{NOM.ACC} licensee because the former does not have a -ACC feature, whereas +CASE{-NOM.-ACC} would be unable to check this licensee.

Currently there are just two predefined paradigms, which are the following.

- (112) 1. Agreement: 1SG, 2SG, 3SG, 1PL, 2PL, 3PL  
 2. Case: NOM, ACC, GEN

There are over 100 selectional properties used in MGbank, and these are shown along with brief descriptions of their purpose in table 3.1. They all have corresponding positive and/or negative selectional requirement features.

+NONE	used to create islands
\$	currency symbol
1SG/2SG/3SG/ 1PL/2PL/3PL	person and number agreement
1/2/3	person agreement

<sup>53</sup>Disjunctive selectional requirements in which all requirements are of the same polarity can be written in Autobank (see the next chapter) with the polarity symbol outside of the brackets, so  $[+\text{NOM}|\text{+GEN}]$  would more likely be written in the following more succinct form in MGbank  $+[\text{NOM}|\text{GEN}]$ .

ACC/GEN/NOM	morphosyntactic case
ADMOD	adverb that modifies other adverbs and adjectives
AGENT	agentive preposition ‘by’
ANA	reflexive/reciprocal anaphor
AS	lexical item ‘as’
ASSOC	DP associate in existentials
AUX	auxiliary
BARE	bare uninflected form of the verb
BE	one of the forms of ‘be’
CASE	marks versions of T which assign case
CLEFT	indicates a cleft clause DP
CMP	comparative
CONJ	conjunct
COORD	negative particle
DAT	null dative head
DECL/IMP/INT/EXCL	illocutionary force
DEF/INDEF	definite/indefinite
DEM	demonstrative
DIR	directional preposition
DIS	discourse level modifier
DO	auxiliary ‘do’
DOU	double object verb
EACH	lexical item ‘each’ in reciprocal ‘each other’
ECHO	wh-in situ item used in echo questions
EDGE	allows a chain to escape SpIC
ELLIP	null head used for ellipsis
EMPH	emphatic ‘do’
EVER	lexical item ‘ever’
EXP	null head triggering movement of associate DP in expletive passives
EXPL	expletive pronoun

EXTRAP	extraposed item
FEM/MASC	feminine/masculine gender
FEW	lexical item ‘few’
FIN/INF	finite/infinitival
FL	associates DP with floating quantifier
FOC/TOP	focalised/topicalised DP
FOR	complementizer ‘for’
GER	gerund
HEAVY	phonetically ‘heavy’ item (e.g. item taking CP complement)
IF	complementizer ‘if’
IND/SUB	indicative/subjunctive
INTRANS/TRANS	unaccusative/passive vs transitive/unergative verb
INV	inverted item
IREPORT	indirect reporting verb
IT	expletive ‘it’
JJR/JJS	comparative/superlative adjectives
LADJ/LADV	light adjective/adverb used in shell structures
LH	lexical head coordinator
LITTLE	lexical item ‘little’
LOC	locative modifier
LV	overt light verb
MAIN/SUBORD	main or subordinate clause
MD	modal
LOWER/MID/UPPER	marks position of null neg heads (for polarity item licensing) in the clausal spine
MNR	manner adverbial
MORE	lexical item ‘more’
NAME	proper noun
NCOMP/NMOD	marks prepositions as capable of being nominal complements/adjuncts

NEG	negative item
NEUT	neuter gender
NOMOD	do not modify
NOT	negative particle
NUM	marks a number noun
ONE	pronoun 'one' used in reciprocal 'one another'
OP	null operator
OVERT	phonetically overt head
NULL	phonetically null head
P	preposition
PART	particle
PASS	passive
PAST/PRES	past/present verb form
PERF/PROG	perfective/progressive verb form
PERS	personal pronoun
PHI	DP with phi features but no epp (for locative inversion)
PL	plural number
PMOD	modifier of PPs
POL	polarity item adverb 'yet'/'ever'
POSS	possessive pronoun
PRD	predicate
PRO	proform
PURP	purposive modifier
RBR/RBS	comparative/superlative adverb
REL	relativised NP
RELAT	relative clause
REPORT	direct reporting verb
RRB/LRB	right/left parentheses
RREL	reduced relative clause
S	possessive 's
SG/PL	singular/plural agreement

SH	shell structure
SO	lexical item ‘so’
SPOSS	strong genitive pronoun
STRANS	semi-transitive verb like ‘say’ taking external argument but not assigning accusative case.
THAN	lexical item ‘than’
THAT	complementizer ‘that’
THERE	expletive ‘there’
TMP	temporal modifier
TO	preposition ‘to’
V	verb
VCOMP/VMOD	preposition able to be verbal complement/modifier
WH	wh item
WHAT	lexical item ‘what’
WHMOD	modifier of wh items
YET	lexical item ‘yet’

Table 3.1: Selectional properties used in MGbank. The +NONE requirement is included because there is no corresponding NONE property in MGbank. This feature is used to create islands, e.g. +top{+NONE} creates a domain that is opaque to topicalisation extraction because no moving item can ever have a NONE property, this being absent from the lexicon.

### 3.4.2 Fine-grained Subcategorization

As well as constraining Move operations, I-selectional restrictions can just as straightforwardly be used to constrain Merge. For instance, we can ensure that a subject control verb like *want* subcategorises for an infinitival CP complement and thereby avoid overgenerating *Jack wants that she help(s)* simply by using the following categories for *want* and *that*:

$$\begin{aligned} \text{want} &:: c\{+INF\}=v\{\text{TRANS}\} \\ \text{that} &:: t\{+FIN\}=c\{\text{DECL.FIN}\} \end{aligned}$$

Because *that* lacks the INF feature required by *want*, the unwanted derivation is blocked. We also need to rule out *\*Jack wants she help(s)*, where the overt complementizer is omitted. As we saw in section 3.2.7, Minimalists assume that finite embedded clauses lacking an overt C head are nevertheless headed by a null C - a silent counterpart of *that*. A complicating factor is that a null complementizer is also assumed to head certain types of infinitival clause, including the embedded *help* clause in *Jack wants [CP to help]*. Given that these null C heads are (trivially) homophones and that they arguably exist to encode the same illocutionary force,<sup>54</sup> an elegant approach would be to minimise the size of the lexicon - and hence the grammar - by treating them as one in the same item. On the other hand, using a single null C head syncretised with both FIN and INF will fail to block *\*Jack wants she help(s)*.

Note that at present both *that* (C) and [pres] (T) are specified as FIN, suggesting a redundancy. Let's therefore assume that T, being the locus of tense, is also the sole locus of inherent finiteness, but that C's selectee may inherit FIN or INF from its TP complement as the derivation proceeds.<sup>55</sup> Only a null C which inherits INF from a *to*-TP complement will be selectable by a verb like *want*, blocking the ungrammatical *\*Jack wants she help(s)*. Note, however, that although lacking inherent tense properties, certain C heads continue to bear inherent tense requirements;<sup>56</sup> for instance, *that*'s selector will retain its inherent +FIN, identifying it as a *finite* complementizer, in contrast to an infinitival complementizer like *for*, whose selected will be associated with +INF.

To implement this percolation mechanism,<sup>57</sup> we now introduce *selectional vari-*

---

<sup>54</sup>Infinitival complementizers are sometimes assumed to encode *irrealis* force (see e.g. Radford (2004)) in contrast to *that* and its null counterpart which encode declarative force. However, the fact that *Jack expects her to help* is (on one reading) virtually synonymous with *Jack expects that she will help* suggests that in both cases the C head is encoding the same semantic property, with any subtle difference in meaning attributable to the contents of T (i.e. *to* vs. *will*). Consider also *Mary wondered whether to help* vs. *Mary wondered whether she should help*, where the embedded infinitival and finite clauses are both clearly interrogative.

<sup>55</sup>Note that if Grimshaw (1991) is correct that functional projections like DP, TP and CP are part of *extended projections* of lexical heads, then we should not be surprised to find instances where fine-grained syntactic properties are projected up through these functional layers.

<sup>56</sup>The property vs. requirement distinction is related to Chomsky's interpretable vs. uninterpretable feature distinction. For example, 3SG on a noun is interpretable, whereas +3SG on a verbal head is uninterpretable.

<sup>57</sup>Note that because we are only allowing selectional properties and requirements to percolate, not the structure building features themselves, this system is fundamentally different from that described in Kobele (2005), where it was shown that allowing licensee features to be percolated leads to type 0 MGs. The present system does not extend the expressive power of MGs beyond that of an MCFG because there is still a finite number of possible MG types at the derivation tree terminals and non-terminals. This is because the lexicon and the set of selectional properties and requirements are both finite, and if a selectional property or requirement percolates onto a non-terminal that already contains an instance

ables, which we write as  $x, y, z$  etc. A variable on a selector or licenser feature will cause all the selectional properties and requirements contained on the selectee or licensee feature that it checks to replace all other instances of that variable on the selecting or licensing category's remaining unchecked feature sequence.

To see how this works, consider the following lexicon.

$$\begin{aligned} [\text{trans}] &:: >v\{+\text{TRANS}.x\} = =d\text{ lv}\{x\} \\ [\text{pres}] &:: \text{lv}\{+\text{PRES}.x\} = +\text{CASE}\{+\text{NOM}.x\} \text{ t}\{\text{FIN}.x\} \\ \text{to} &:: \text{lv}\{+\text{BARE}.x\} = \text{t}\{\text{INF}.x\} \\ [\text{decl}] &:: \text{t}\{x\} = \text{c}\{\text{DECL}.x\} \\ \text{that} &:: \text{t}\{+\text{FIN}.x\} = \text{c}\{\text{DECL}.x\} \end{aligned}$$

Observe that the [pres] T head has an  $x$  variable on its selector feature and that this same variable also appears on its licenser and selectee; any selectional properties or requirements contained on the lv selectee of its vP complement will thus percolate onto and replace these two variable features (see fig 3.41). Note also the FIN property on [pres]; the  $x$ 's on the two C heads will percolate this property to the c selectee, where it can be selected for by a verb like *say*, but not *want*, which requires INF (from *to*); this will correctly block *\*Jack wants (that) she help(s)*.

$$\begin{array}{l} \varepsilon, \varepsilon, \text{helps him} : +\text{CASE}\{+\text{NOM.PRES.TRANS.+3SG}\} \text{ t}\{\text{FIN.PRES.TRANS.+3SG}\}, \text{he} : -\text{case}\{\text{NOM.3SG}\} \\ \varepsilon, [\text{pres}], \varepsilon : \text{lv}\{+\text{PRES}.x\} = +\text{CASE}\{+\text{NOM}.x\} \text{ t}\{\text{FIN}.x\} \quad \varepsilon, \text{helps, him} : \text{lv}\{\text{PRES.TRANS.+3SG}\}, \text{he} : -\text{case}\{\text{NOM.3SG}\} \end{array}$$

Figure 3.41: Merger of T with vP results in the percolation of selectional properties and requirements from v's selectee feature (lv) to T's licenser (+CASE) and selectee (t) features.

MGBank also uses this same technique to enforce the long-distance subcategorization of subjunctive verb forms, as illustrated by 113 below.

(113) He demanded that we be/\*are there on time.

of it, the parser simply unifies the two instances. Thus there can never be, for example, two instances of the ACC feature associated with the same structure building feature. Because the set of MG types at the terminals and non-terminals of all possible derivation trees is finite, we could still convert our MG into a strongly equivalent LCFRS/MCFG, as described in Mainguy (2010). Of course, given that there are well over 200 selectional property and requirement features used in MGBank, the set of possible non-terminal MG types is very large and hence any MCFG converted from this grammar would also be very large (the ordering of the selectional properties and requirements on each structure building feature is at least fixed).



In standard English, certain verbs (such as *demand*, *insist*, *require* etc) selecting for CP complements require that the verbs embedded inside those CP complements be in their bare, uninflected form. This is an interesting phenomenon because it appears as if what we have here is a case of selection at a distance. MGBank enforces this selectional restriction for these verbs by marking the v selectee of uninflected verb forms with the property BARE and introducing the following special subjunctive T head, which selects for a vP whose selectee has the property BARE (in the lexicon this property resides on the main verb but during the derivation it will percolate from the v selectee of the verb onto the lv selectee of the little v head).

$$[\text{sub}] :: \text{lv}\{+\text{BARE}.x\} = +\text{CASE}\{+\text{NOM}.x\} \text{t}\{\text{FIN.SUB}.x\}$$

The SUB property of this T head's selectee will percolate from the t selectee of TP to the c selectee of CP, where it can be selected for by a verb like *demand*. To ensure that other types of verbs, like *say*, avoid selecting a subjunctive CP complement, MGBank includes a negative selectional requirement -SUB on the c= selector of these verbs (+IND could just as easily be used, of course).

### 3.4.3 Agreement

As noted, the MGBank grammar follows Chomsky (1995) in assuming that all lexical items enter the derivation fully valued for case and agreement features. However, in MP subjects enter into agreement relations with the T head when they move to their surface spec-TP position, rather than with the inflected verb directly, so something more needs to be said in order to enforce the mediated subject-verb agreement and avoid overgenerating, e.g. *\*he help him*.

One (not very elegant) solution would be to use multiple T heads for each person and number combination. For instance, 3SG T could select 3SG vP (vP having inherited 3SG onto its selectee from VP's selectee) and case license only a 3SG subject.

MGBank avoids such lexical duplication, however, by instead placing agreement requirements such as +3SG, +1PL, -3SG etc on the v *selectee* feature of the lexical verb. These then percolate up to the +CASE licenser of the T head where they specify the corresponding agreement properties on the subject's -case licensee. We thus have the following updated entries.

$$\text{him} :: \text{d -case}\{\text{ACC.3SG}\}$$

he :: d -case{NOM.3SG}  
 helps :: d= +CASE{+ACC} v{+3SG.PRES}

Recall that an  $x$  variable was included on both the  $t$  selectee *and* the +CASE licensor of  $T$ ; this ensures that any selectional requirements (and properties) of the main verb<sup>58</sup> (crucially here the +3SG of *helps*) are percolated onto  $T$ 's +CASE licensor (via  $vP$ ), where they will enforce the correct subject-verb agreement (see fig 3.41). Note too that tense agreement between the  $T$  head and the inflected lexical verb is also enforced by the +PRES requirement on  $T$ 's  $lv=$  selector, the PRES property having percolated up from the lexical verb to the transitive light verb's  $lv$  selectee. The treatment of the relation between  $T$  and  $V$  as one of agreement rather than as involving affix hopping makes this approach somewhat akin to the approach taken in Roberts (2010).

### 3.5 Related work on agreement and fine-grained subcategorisation

The idea of using fine-grained selectional features to constrain the grammar is certainly not new. The CCGbank grammar, for example, uses a limited inventory of such features to sub-classify  $S$ ,  $NP$  and  $N$  nodes, e.g. as  $S[dc]$ ,  $S[to]$ ,  $S[q]$ ,  $NP[nb]$ ,  $N[conj]$ , and so on. Agreement is not handled in this way, however, but is instead left to the statistical model. The approach to agreement and fine-grained subcategorisation proposed here is similar to that which was implemented in a transformational grammar context for the Syntactica project (Larson et al., 1996; Larson, 2009). Syntactica was designed as an educational tool to enable students to construct transformational grammars, and is in some ways similar to the Autobank system described in the next chapter. The grammar it uses includes features such as +TNS and -TNS, which have the interpretation 'is tensed' and 'is not tensed' respectively (equivalent to MGBank's FIN and INF). These features can be included in the subcategorisation frame of a given head. For example, the complementizers *that*, *for* and *whether* are given the following lexical entries in Larson (2009):

that, C, [ $+\_TP_{+TNS}$ ]  
 for, C, [ $+\_TP_{-TNS}$ ]

<sup>58</sup>Note that selectional requirements are entirely inert while located on selectee features; conversely, selectional properties are inert on selectors and licensors.

whether, C, [ $_{-}$ TP $_{+TNS}$ ], [ $_{-}$ TP $_{-TNS}$ ]

The +TNS associated with the TP complement in the subcategorisation frame of *that* ensures that this complementizer only selects for a finite TP complement and conversely for the -TNS on *for*. In the MG formalism, subcategorisation frame are encoded by the structure building feature sequences. MGbank therefore encodes the same information as above for *that* and *for* by associating +FIN or +INF features with the  $t=$  selectors of these complementizers (-FIN could just as easily have been used as +INF, but the negative selectional requirements were only introduced later, after INF was already part of the grammar). The interrogative complementizer *whether*, meanwhile, has a disjunctive subcategorisation frame above to allow it to take either a finite or non-finite clause complement; in MGbank, such disjunction could be expressed via the feature +[FIN|INF], but since FIN and INF are two halves of a binary finiteness paradigm, the same effect is achieved simply by not adding any finiteness requirement to the  $t=$  feature of *whether*.

The Syntactica grammar also allows for the percolation of features from heads to phrases. MGbank does not only allow percolation from the heads of phrases, however, but also from non-heads; in section 3.4.2, we saw how this mechanism can be useful for capturing long-distance subcategorisation in English subjunctives because it allows the SUB feature to percolate from TP to CP where it can be selected for by the embedding verb, for example. Free relative clauses appear to be a clear example of where specifier dependents are able to project (see section 4.2.1 below). For example, in the minimal pair 114 and 115 below, whether or not the modifying clause is categorised as a temporal or locative adjunct is clearly determined by the wh-specifier in the left periphery of each clause.

(114) I will go when you go.

(115) I will go where you go.

It must however be conceded that the feature system proposed here as it stands is not sufficiently constrained, as it does allow for feature percolations which are clearly not found in natural language. For example, there would be nothing stopping the grammar designer from introducing a verb which inherits the finiteness property of its CP complement. And even in the free relative case, it is clear that while some of the wh specifier's features, such as its nominal vs adverbial, or locative vs temporal features, may project, others, such as number, do not (*which books you read doesn't/\*don't*

*interest me*).

The dot notation used here for separating multiple fine-grained subcategorisation properties (e.g. NOM.ACC.3SG.FEM) is also found in the context-free grammar of the Gramotron German parser (Beil et al., 1999; im Walde et al., 2001; Beil et al., 2002) which was based on the feature-based grammar of Schmid (2000) (although in Schmid (2000) semi-colons are used instead, with dots reserved for indicating structured feature-value paths, e.g. Agr.Number=sg). For example, Beil et al. (2002) represent the category for the German word *bleibe* (meaning ‘residence’ as a noun, or ‘stay’ as a verb) as follows:

Bleibe {NN.Fem.Cas.Sg, VVFIN}

The curly braces are used to enclose multiple morphosyntactic representations of a single word form, which are comma separated. We will focus here on the nominal category to the left of the comma, which contains the dot separated features that are properties of this noun: Fem refers to feminine gender and Sg to singular number, while Cas is a four-way disjunctive case feature standing for nominative, accusative, genitive or dative cases; disjunctive features appear lower down in the tree and are resolved higher up. For example, fig 3.42 shows the Gramotron tree for the German phrase *eine gute Gelegenheit* (‘a good opportunity’). At the preterminal level, the noun is represented by the same feature sequence as that shown above for *Bleibe*, but by the next node up the four-way disjunctive Cas feature has been replaced by the two-way disjunctive Dir feature which stands for nominative or accusative. The reason for this is that in German, adjectives agree in case with the nouns they modify and the *-e* suffix on *gute* indicates nominative or accusative (but not genitive or dative) agreement. German articles also agree in case, but the *eine* form of the indefinite article can also be used for both nominative and accusative agreement, and so at the root of this tree are two possible morphosyntactic representations for this phrase, each of which has an unambiguous Nom or Akk (accusative) feature.

Both the Syntactica and Gramotron grammars use re-write rules, with the percolation of specific features hard-coded onto these rules. This contrasts with the MGBank grammar, which is strongly lexicalised with only a few very abstract rule schema, meaning that the reified percolation operations must be specified on the lexical categories themselves. One drawback to the MG feature system described so far is that it is not possible to selectively percolate certain features. For example, notice that in fig

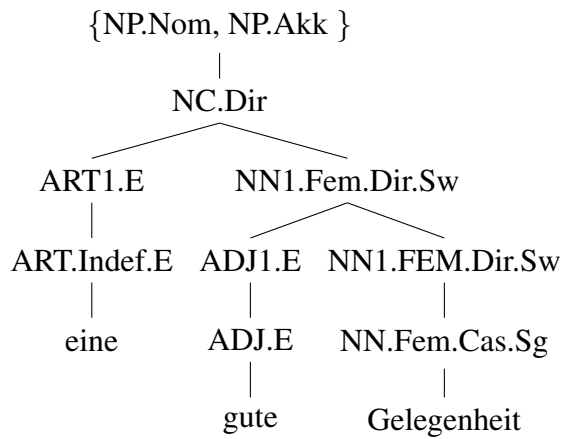


Figure 3.42: A Gramotron-style tree for the German phrase *eine gute Gelegenheit* ('a good opportunity').

3.42 the Fem feature is percolated up to the NN1 level whereas the Sg feature doesn't advance higher than the preterminal level (the Gramotron grammar only enforces number agreement within nominals, not between subjects and verbs, and adjectives have the same ending in the feminine singular as they do in the plural, while the indefinite article has no plural form). Section 4.4.1 below introduces selectional suppressor features which can be used to achieve this effect, although it must be conceded that this is not a particularly elegant solution.

A more faithful valuation-based version of Agree for MGs that is inspired by the feature-sharing approach of Frampton and Gutmann (2000) has recently been proposed in Ermolaeva (2017). This proposal was made well into the development of the MGbank corpus, hence it was not possible to integrate it with MGbank, though there is nothing to prevent this system from being retrofitted to the corpus in the future. More complex (unbounded) nested and typed feature structures are employed alongside unification operations in a number of formalisms, most notably HPSG (where they are used not just to capture local agreements but also long-distance dependencies). However, the system proposed here, although less linguistically sophisticated than these other approaches, does have the important practical advantage that its notation is very simple and hence easy to annotate at speed, and this greatly facilitated the development of the manually annotated portion of MGbank.

# **Chapter 4**

## **Locality**

## 4.1 Introduction

Sections 3.2.6 and 3.2.10 introduced Stabler’s Shortest Move Constraint, which is a strict version of the intervention effects often assumed in mainstream Minimalism. A number of other locality constraints from the linguistics literature were also briefly mentioned, including specifier, adjunct and *wh*-islands. These types of constraints have been a central concern within TG since the publication of Ross (1967) and following the introduction of the extremely powerful rule *move- $\alpha$*  in GB. For this reason, this chapter provides an in depth account of how many of the constraints often assumed in mainstream TG have been implemented into MGbank with the aim of keeping movement as constrained as possible.

## 4.2 Derelativized SMC (DSMC)

MGs without head movement can be parsed in  $n^{2k+3}$  time (Fowlie and Koller, 2017), while MGs with head movement have a worst case time complexity of  $n^{2k+5}$  (Stanojević, 2019), where  $k$  is the size of the set of licensee features which are capable of triggering overt movement. It is Stabler’s strict version of the SMC which enables this polynomial time parsing result.<sup>1</sup> However, in the grammars proposed in the mainstream Minimalist literature, there are typically quite a few different overt movement types: in the *A'* domain, constituents can move to check at least topic, focus and/or *wh* features, and some Minimalists also assume separate scrambling and/or rightward movement operations to exist (without rightward movement additional and often unspecified types of leftward movement are required); in the *A*-movement domain, the canonical type of movement is that which checks case/agreement features, but under movement-based approaches to control and binding such as Hornstein (2001), there is also theta-driven *A*-movement. MGbank includes all of these types of movements along with a few others; for example, as already noted, polarity item licensing is treated here as an instance of covert movement, and there are also other types of movement used to constrain operations such as reflexive binding and locative inversion.

With so many different types of licensee,  $k$  becomes very large, and although the asymptotic complexities noted above are only worst case theoretical measures, they can nevertheless have some impact on practical runtimes. In order to ameliorate this

---

<sup>1</sup>This is because there is no need to specify both the mover and the target of movement as arguments to *Move*, only the expression itself need be specified; this enables movement to be represented as a unary branching node in the MG derivation tree, meaning that the set of such trees is regular.

effect, therefore, while still allowing for the full range of different types of movement from Minimalist theory, a stricter, partially derelativized version of Stabler's SMC, called DSMC, is enforced by the parsers constructed for this project. Stabler's traditional SMC is also retained, for reasons which will become clear as the discussion proceeds. Deferring until section 4.2.6 the question of how to handle the -pers, -num, -epp and -loc features which were introduced in section 3.3.5 for existential constructions, DSMC can be stated as follows.

**The derelativized Shortest Move Constraint (DSMC)** (first version: for an EDMG not using the licensees: -pers, -num, -epp or -loc): Two licensee features may both be active at the same time in the derivation only if they are drawn from different feature classes, where the feature classes are: A-movement-1 features (-case, D, -tough), A-movement-2 features (-self), A'-movement-1 features (-top, -foc, -wh) and A'-movement-2 features ( $v\sim$ ,  $t\sim$ ,  $c\sim$ , -n).

DSMC ensures that despite there being (in this version) 11 licensee features capable of triggering overt movement in the grammar,<sup>2</sup> there can only ever be 4 overtly moving chains in the derivation at any one time. Since  $k$  is intended precisely as a measure of the number of non-empty substrings which any constituent can have, this limits  $k$  in the above complexity measures.

Stabler's original SMC is also enforced alongside DSMC for a number of reasons to be discussed in what follows. The first of these is that it covers the two licensees -negs and -pol which are not included in the feature classes identified in the definition of DSMC given above. These licensees are restricted to only triggering covert movement in the MGBank grammar (see section 4.2.5), hence they will not have any impact on the asymptotic time complexity of the parser, provided that the maximum number of each of these licensees active in the derivation at any one time is constrained to some constant number. Classical SMC fixes that constant at one.

Of course, the grammar could be made more efficient if there were only one A-movement class and one A'-movement class. Unfortunately, this did not prove to be feasible. The reason for including two classes of A-movement features in the above definition is that the -self feature involved in the binding of reflexive and reciprocal anaphors must be permitted to co-exist alongside D, as will be discussed in section 4.2.4. Two A'-movement groups are also required, partly in order to allow rightward

---

<sup>2</sup>See section 4.2.5 on overt-only and covert-only movers.



movement to co-occur with leftward A'-movement. For example, in 116 below there is wh movement of *what* out from the embedded clause to the left periphery of the matrix clause, while the complement clause itself must undergo rightward movement in order to be positioned to the right of the temporal adjunct *last night*.

(116) What<sub>i</sub> did Jack promise t<sub>j</sub> last night [that he would do t<sub>i</sub> today]<sub>j</sub>?

A further reason for there being two classes of A'-movement licenses is that MG-bank implements a version of the promotion analysis of restrictive relative clauses proposed in Bhatt (2002), which requires the relativized NP and the wh head to move separately to the left periphery. This necessitates there being two separate licensees for relativisation, -n and -wh, which are simultaneously active in the derivation. We will now make a brief excursus in order to look at the details of this analysis as well as the motivation for it.

#### 4.2.1 An excursus on restrictive relative clauses

There is some controversy within TG over the best way to analyse restrictive relative clauses, such as the object relative clause in 117 below.

(117) the [book of ghost stories which Jack read] was great.

We can broadly identify two main approaches, which Bhatt (2002) refers to respectively as the *head external* analysis and the *head raising* analysis<sup>3</sup> (also known as the *head promotion* analysis). Bhatt (2002) notes that the origins of the ubiquitous head external analysis are unclear, but that Quine (1960) seems to suggest it and that it is assumed in Montague (1970), Partee (1975), Chomsky (1977) and Jackendoff (1977); it is also the analysis presented in Radford (2004). Under this analysis, the head NP is generated outside of the relative clause, while the wh operator is generated inside the relative clause and then undergoes A'-movement to its left periphery. The relative clause is adjoined to the head NP and combines with it under intersective modification. This larger NP is then selected as the complement of *the*. The schematic structure for the relative clause in 117 under the head external analysis is shown in 119 below.

(118) [<sub>DP</sub> the [<sub>NP</sub> [<sub>NP</sub> book of ghost stories] [<sub>CP</sub> which<sub>i</sub> Jack read t<sub>i</sub>]]]

<sup>3</sup>A third, *matching* analysis, which is discussed in Bhatt (2002) and is half way between these two will be left aside here.

The alternative promotion analysis has been proposed in various forms by a number of different authors (Brame (1968), Schachter (1973), Vergnaud (1974), Åfarli (1994), Kayne (1994) and Bhatt (2002)). Kayne's (1994) analysis has been particularly influential. Under this proposal, the head NP starts out inside the relative clause as the complement of the *wh* operator, before moving to the latter's spec, with this DP subsequently undergoing movement to the left periphery. The CP clause is then selected as the complement of the determiner *the*. This analysis is shown in schematic form below.

(119) [<sub>DP</sub> the [<sub>CP</sub> [<sub>DP</sub> book of ghost stories which]<sub>i</sub> Jack read t<sub>i</sub>]]

For arguments in favour of head promotional analyses, the reader is referred to Bhatt (2002). Here we will look at just one empirical argument in its favour which seems to me particularly persuasive. In section 3.2.7, we saw that one of the standard tests for constituenthood used in TG is idiomaticity. We also noted in that section that what seems to matter for this test is not surface structure, but deep structure; this makes sense if idioms are stored in the lexicon as chunks of structure, which would explain why they exhibit a non-compositional lexical semantics. In view of this, consider the following examples from Schachter (1973) (cited in Bhatt (2002)).

- (120) a. We made headway.  
 b. \*(The) headway was satisfactory.  
 c. The headway that we made was satisfactory.

Example 120a exemplifies the idiom to *make headway*. Example 120b shows that it is not possible to use *headway* on its own in this type of context, i.e. without *make*. In view of this, the acceptability of 120c is arguably surprising under a head external analysis in which at no point in the derivation does *headway* form a constituent with *make*. Under the head promotion analysis, on the other hand, the idiomatic meaning is preserved because *made headway* does indeed begin the derivation as a constituent, with *headway* (and in this example a null *wh* operator) subsequently moving to the left periphery of the relative clause.

The promotion analysis is also potentially attractive from a statistical parsing perspective because by capturing the dependency between the head NP and the verb within the syntax (rather than establishing it in the semantics via intersective modification), we enable the statistical model to condition over this dependency.

Bhatt (2002) proposes an interesting variant of Kayne's analysis, a version of which is implemented in MGBank. As in Kayne's analysis, in Bhatt's analysis the head noun begins the derivation by being merged as the complement of the wh operator inside the relative clause before both items move to the left periphery of that clause. Where the two analyses diverge is that only in Kayne's analysis does the NP move to the specifier of the DP headed by the wh operator.<sup>4</sup> In Bhatt's (2002, page 81) analysis, on the other hand, once the wh-NP complex moves to spec-CP, the NP breaks away from its complement position and moves to the specifier of a higher (null) nominal head governing the CP; the determiner then selects this NP shell layer as its complement, rather than selecting the CP directly. An important aspect of this proposal is that unlike in Kayne's analysis, the wh operator and the NP do not form a constituent in the surface structure here; instead, the wh constituent forms a constituent with the rest of the relative clause. These two different promotion analyses are shown in fig 4.1 for the phrase *the book of ghost stories which Jack read*.

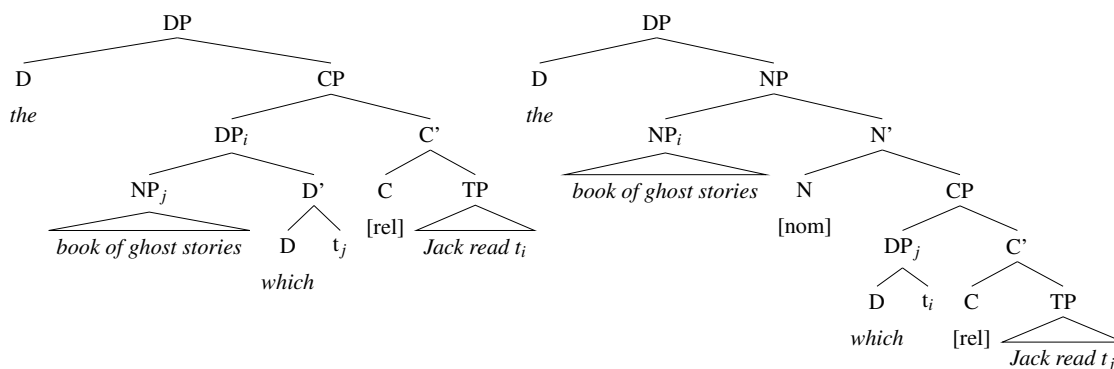


Figure 4.1: The promotion analyses of Kayne (1994) (left) and Bhatt (2002) (right) for restrictive relative clauses in English. In both analyses the wh operator and head NP form a constituent early on the derivation, but only in Kayne's analysis do they form a constituent in the surface structure.

One reason for preferring Bhatt's analysis over Kayne's is that the wh operator does indeed appear to form a constituent with the rest of the relative clause which excludes the head noun, as evidenced by the fact that they can be coordinated in 121.

(121) The books [which Jack bought] and [which Pete read].

<sup>4</sup>It is unclear to me what the (semantic) motivation for this movement could be, other than to derive the correct word order.

Bhatt does not specify why the NP moves to the specifier of the higher NP or which feature(s) drive this movement. From a deep competence-based perspective, one possibility is that there has to be a nominal layer in order for the relative clause to be selected as the complement of *the* (just as a standard *that* clause can only be selected by *the* if the dummy noun *fact* is inserted between the determiner and the complementizer, e.g. in *the \*(fact) that people said that...*). If we further assume that the null nominal head is defective and must be combined with overt nominal material in order to successfully project an NP layer, then it could be that the head NP moves to the spec of this defective nominal head in order to meet these requirements.

Notice that this implies that a specifier can project, instead of, or as well as, the item which selects/attracts it. Chomsky (2008, 2013) (building on work by Cann (1999)) has recently argued precisely that specifiers can indeed project (Bhatt also considers an alternative analysis for restrictive relative clauses under which the NP adjoins to the CP and projects over it). In fact, relative clauses more generally appear to provide evidence for specifier-projection (Iatridou et al., 2001). Notice, for example, that a free relative clause behaves as a nominal if the *wh*-operator in its spec-CP position is a nominal (e.g. it can be the object of a transitive verb (122)) and as an adverbial if the *wh*-operator is an adverbial (e.g. it can modify an already fully saturated verb (123)).

(122) It devours [whatever it sees].

(123) He'll read it [when he has some free time].

This suggests that the specifier of CP in free relatives may optionally project, rather than the C head. As (Chomsky, 2008, page 12) observes, if instead the C head itself projects, then in place of a free relative we find an embedded *wh*-interrogative.

(124) I wonder [when he'll have some free time].

In section 3.4.2, we saw that the fine-grained selectional properties and requirements (NOM, FIN, +3SG, -INF etc) can percolate up the tree from the selectee/licensee of a dependent to the selectee/licensee, of the selecting head. We therefore already have a mechanism in the present formalism by which specifiers can project these fine-grained properties. What about the structure building features themselves? Can the relativised noun also project its *n* feature to the defective NP layer dominating the CP in a Bhatt-style analysis? Unfortunately, Kobele (2005) has shown that allowing specifiers to project structure building features results in type-0 MGs. The MG formalism presented here therefore does not allow for this type of pied-piping mechanism.

Fortunately, we can achieve the same intended effect here without increasing expressive power by simply precompiling the percolation directly into the lexicon. To do this, we will add to the lexicon a null [nom] head which both attracts an NP with a -n licensee to move to its specifier and already has an n selectee behind its +N licenser; any fine grained selectional properties on the -n of the relativised NP, meanwhile, will percolate to the n of the [nom] head. The [nom] head will select a CP complement headed by a [rel] C head, which in turn will select a standard declarative CP complement (this allows MGBank to treat relative *that* and complementizer *that* as one in the same item). These three null heads are shown below.

$$\begin{aligned} [\text{relativizer}] &:: n\{x\}=n\{x.\text{REL}\} -n\{x\} \\ [\text{rel}] &:: c\{+\text{DECL}\}=+\text{WH } c\{\text{RELAT}\} \\ [\text{nom}] &:: c\{+\text{RELAT}\}=+\text{N}\{x\} n\{x\} \end{aligned}$$

The [relativizer] head will select an NP, effectively pied-piping its n feature and adding a new relativisation licensee -n to it. Any selectional properties (such as MASC, 3PL etc) will also be pied-piped onto the n and -n features of the resulting constituent owing to the *x* variables associated with all three of the structure building features of the [relativizer] head. This compound [relativizer]+NP constituent will then be selected by the wh-determiner/operator, but will immediately break away from it because it has an active -n licensee in need of checking. The wh operator will also have its own active -wh feature to check. Subsequently, after the inner [rel] CP has been constructed and the wh operator has moved to become its specifier, the [nom] head will select this CP and its +N licenser will attract the relativized NP to become its specifier, placing it to the left of the wh operator. The *x* variable on the +N and n features will mean that the relativised NP will project its fine-grained selectional properties to the larger NP. Notice that in this case there is no variable appearing on the *c*= selector, meaning that the CP will not project its own selectional properties here.<sup>5</sup>

The MGBank-style derivation tree for the phrase *book of ghost stories which Jack read* corresponding to the NP relative clause NP in fig 4.1 is given in fig 4.2. This NP can be selected for straightforwardly by any standard determiner, which is another advantage of adopting Bhatt's analysis over Kayne's, as in the latter case we would

<sup>5</sup>Note that it would also be possible to allow *both* the CP and the NP to project their fine-grained selectional properties; this could be achieved using the following category for [nom], which features both *x* and *y* variables encoding each dependent's projection.

$$[\text{nom}] :: c\{x.\text{RELAT}\}=+\text{N}\{y\} n\{x.y\}$$

have needed to add to the lexicon a new set of construction-specific determiners which select for CP, rather than NP, complements.

The reader may at this point be wondering about examples, such as 125 below, which do not include an overt wh-operator.

(125) The book (that) Jack read.

The MGBank lexicon includes null wh operators which play virtually the same role as the overt one in the derivation in fig 4.2.<sup>6</sup> In MGBank, *that* is never treated as a relative pronoun (which is in accordance with standard Minimalist assumptions - see Radford (2004)), but always as a complementizer head, and so for this example it would occupy the slot filled by the null [decl] head in fig 4.2. See Appendix A for the lexical categories used in other types of relative clause, including appositive, free, reduced, infinitival and restrictive adverbial relative clauses. See also section 4.4.1.2 for how MGBank blocks the derivation of examples such as *\*the book which that Jack read*, which violate the so-called *Multiply Filled Comp Filter*.

Returning to the relevance of restrictive relative clauses for DSMC, notice that in the derivation in 4.2, the features -wh and -n are simultaneously active. This means that they must be kept in separate feature classes or they will trigger a DSMC violation, which is another reason why two A'-movement classes were included. Since relativisation will often remove the need for heavy DP shift, the fact that -n is in the same feature class as the rightward movement licensees is not too problematic as far as corpus coverage is concerned. There are, however, certain sentences which the grammar cannot generate as a result of this strategy, such 126 below, which would require rightward movement and -n licensees to be active at the same time.

(126) The man to whom Pete gave  $t_i$  yesterday [that book of ghost stories by Susan Hill] $_i$ .

<sup>6</sup>The only difference is that in these cases the null wh operator and the relativized NP remain as a single, overt, constituent throughout the derivation, with the null wh operator 'pied-piping' the -n licensee of the relativised NP in the usual non-literal sense of this being precompiled into its lexical entry. This needs to occur here because the movement to check the +WH licenser of the [rel] C head must be overt, and the present formalism assumes that any moving chain which is null is automatically moving covertly, irrespective of whether it began its derivational life as an overt constituent. Of course, in the case of null wh operators we do not have to worry about the constituency facts which we observed in example 121 above. The (simplified) type of the null [wh] head is given below.

$$n\{+REL.x\} = +N\{y\} D\{x\} -case\{x\} -wh\{x\} -n\{y\}$$

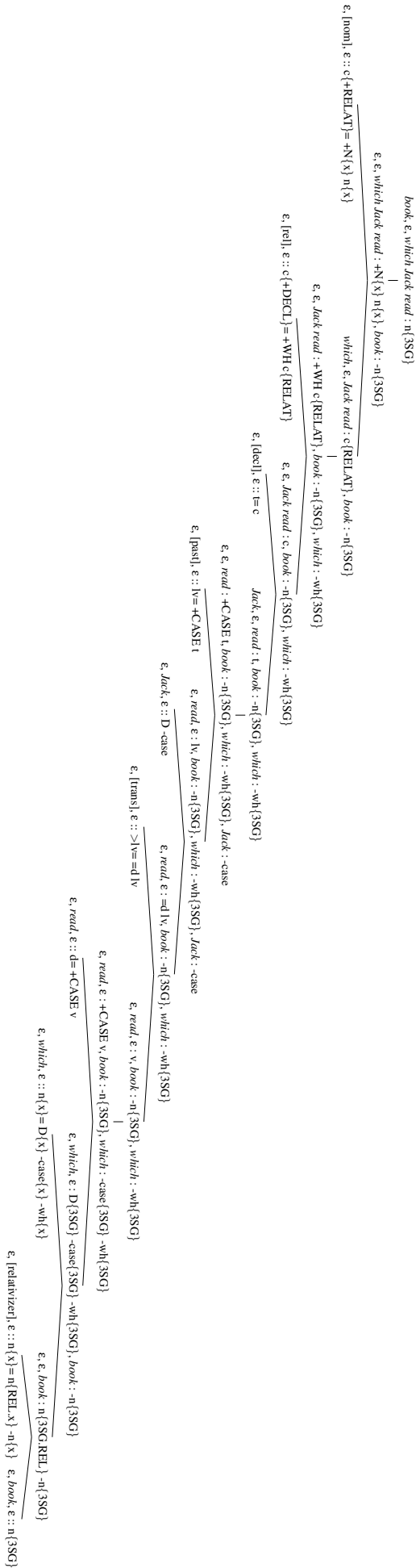


Figure 4.2: A derivation tree for the bracketed NP in the [book which Jack read] (selectional properties and requirements irrelevant to the discussion at hand are omitted to save space on the page).

### 4.2.2 Empirical motivation and problems for DSMC

DSMC was introduced primarily in order to improve the efficiency of parsing. It is inspired by Rizzi's (1990) original formulation of Relativized Minimality. Consider again the configuration from section 3.2.10, repeated below.

(127)                                    [... A ... [... B ... [... C ... ] ] ]

In Rizzi (1990), if both A and B are potential antecedent governors for C, B will block government of C by A just in case A and B are both governors of the same type, where Rizzi defined types as being head governors and antecedent governors, the latter breaking down into antecedent governors in A'-positions and antecedent governors in A positions. So an A-governor could only block an A'-governor and vice versa. In the present context, the highly local nature of head movement is built into the rules in section 3.3.2, although we also saw exceptions in the form of excorporation in section 3.3.7.4.<sup>7</sup>

The distribution of features across the four classes for DSMC was chosen in such a way as to try to minimise for undergeneration and overgeneration, although of course we cannot hope to capture everything given the relatively coarse system of feature classes we are working with; and in any case, judgements of sentence acceptability with respect to locality constraint violation are often subtle, gradient and variable between speakers.

An arguable example of the successful avoidance of overgeneration by DSMC is that so-called topic islands are blocked. This is because they would have to involve two active licensees from the class A'-movement-1 in the derivation at the same time. Haegeman (2012) provides the following examples, arguing that they show that a topic in an intermediate spec-CP blocks long wh extraction (128), focalization (129), topicalization (130) (also handled by standard SMC), and relativization (131).

- (128) a. You said that to Sue Bill introduced Pete.  
       b. \*Who did you say that to Sue Bill introduced? (Boeckx and Jeong, 2004, page 84)

---

<sup>7</sup>Rizzi (2004) subsequently refined his classification so that the monolithic A'-movement class was decomposed into three separate modifier, quantificational, and topic classes. These three classes are unrelated to the two classes in our definition of DSMC, however. Rizzi, of course, was not concerned as we are with keeping *k* as low as possible.



- (129) a. I think that these books they will give to John.  
 b. \*To JOHN I think that these books they will give.

- (130) a. \*To John I think that these books, they will give.

- (131) a. \*This is the book which I think that to John they gave.

On the other hand, Haegeman (2012) also presents data which appear to show that in other cases the features from the A'-movement-1 class can cross one another without problem. For example, 132a features extraction of a topic across a wh-constituent, while in 133a we even have extraction of a relative wh constituent across an interrogative wh constituent (the latter being arguably a violation even of standard SMC).

- (132) a. This book, I was wondering who might be interested in reading.

- (133) a. This is a problem which I am not sure how to solve.

The MGBank grammar also currently cannot generate examples such as 134 below, in which there is topicalization (of *that kind of behaviour*) and focalization (of *never again*) within the same clause. This again violates DSMC because the -top and -foc licensees involved are of the same A'-movement-1 class and would have to be present in the derivation at the same time.

- (134) That kind of behaviour never again will I tolerate.

Most of these examples are outliers which one would only encounter in linguistics textbooks, of course. From a practical perspective, therefore, not generating them is unlikely to have any impact on parser performance, and so the efficiency gains should comfortably outweigh the ensuing cost in terms of recall.

In the A-movement domain, DSMC turns out to have some very welcome effects, along with a few complications to which we turn in the following sections. Boeckx (2010) represents the most fully worked out theory of control as A-movement, and it seems clear from the discussion in sections 4.4.2 and 5.2.1 of that work, that these authors regard all thematic DPs as potential interveners for both case- (or phi-) driven A-movement and theta-driven A-movement. This is understandable given that mainstream Minimalist analyses generally do not assume features on lexical items to be ordered. Therefore, a DP which is active in the derivation arguably has all of its features/properties visible to the computation. Since (at least all argumental) DPs have

both case and theta properties,<sup>8</sup> all DPs are potential interveners for all other DPs for both types of A-movement, at least while the relevant features remain unchecked. DSMC approximates this perspective by placing -case and D licensees in the same feature class (see section 3.3.6 on control and D as a theta licensee).

We saw in section 3.2.10 that in order to avoid unwanted SMC violations, the MG-bank grammar had to be designed in such a way as to avoid either crossing or nested case-driven A-movement dependencies. This approach must also be extended to include *all* A-movement-1 dependencies, i.e. both case- and theta-driven A-movement, in order to avoid DSMC violations; in the majority of cases, it turns out to be straightforward to ensure that every DP is both case licensed and theta licensed before another active DP is introduced into the structure. For example, consider 135 below.

(135) Jack<sub>i</sub> seemed to Pete to t<sub>i</sub> help Mary.

By the time that *Jack* is merged into spec-vP of the *help* clause, *Mary* has already checked and deleted both her D and -case features against the d= and +CASE features of *help*. Furthermore, although in linear terms the DP Pete intervenes between *Jack*'s surface position and his trace in spec-vP of the lowest clause, this DP is governed by a preposition, hence it will have had its D and -case features checked and deleted before the PP is merged into the structure. On the other hand, an intervening bare DP does block control by the higher DP, as expected.

(136) Jack<sub>i</sub> [<sub>vP</sub> t<sub>i</sub> expected Mary<sub>j</sub> to [<sub>vP</sub> t<sub>j/\*i</sub> try to [<sub>vP</sub> t<sub>j/\*i</sub> help.]]]

In this example, only *Mary* can be doing the helping and the trying, not *Jack*, because *Mary* intervenes between *Jack* and the lower verbs. However, we shall now see that this intervention implies that theta-driven A-movement must be allowed to trigger intervention effects for case-driven A-movement in the present framework. In other words, DSMC is required over and above SMC in order to correctly block the co-indexation indicated in 137 below (this overgeneration was detected when the Autobank parser returned structures with the incorrect indexation indicated below, which highlights the utility of computational testing).

(137) Jack<sub>i</sub> [<sub>vP</sub> t<sub>i</sub> expected Mary<sub>j</sub> to [<sub>vP</sub> t<sub>i</sub> try to [<sub>vP</sub> t<sub>j</sub> help.]]]

<sup>8</sup>I hesitate to use the word 'features' because although Boeckx et al. (2010) refer to phi-features they do not actually use the term 'theta features'. Instead, they refer to A-movement for 'theta reasons' (e.g. page 129). In an MG context, however, this movement must be fully formalised, and theta-driven movement is therefore treated here as being driven by the need to check and delete features, just as all other movement is.

MGbank adopts the *raising-to-object* (or rather, raising-to-spec-VP) analysis of ECM verbs such as *expect* that is suggested in Chomsky (2008) (the spirit of which dates back to Postal (1974)).<sup>9</sup> Under this analysis, *her* in 136 raises out of the *try* clause to spec-VP of *expect* to check -case (after which *expect* undergoes standard V-to-v head movement, placing it to the left of *her*). Crucially, as we saw in section 3.2.7, unlike in the case of superficially similar object control verbs such as *persuade*, ECM verbs clearly do not assign a theta role to their DP objects, but merely check their -case feature (as we also saw in section 3.2.7, this semantic distinction has a one-to-one correspondence with the different syntactic types used for these two classes of verbs). Thus while 138a entails 138b, 139a does not entail 139b.

- (138) a. Jack persuaded Mary to help.  
       b. Mary was persuaded (by Jack.)
- (139) a. Jack expected Mary to help.  
       b. Mary was expected (by Jack).

A derivation for 136 which yields the correct argument dependencies is given in fig 4.3. The important points to note are that *Mary* checks and deletes the =d feature of the *help* clause via External Merge (picking up her first AGENT theta role), after which she also checks the =d feature of the *try* clause via control movement (picking up her second AGENT theta role), before moving to her final surface position in spec-VP of *expect* to check -case. Only after all of this has taken place is *Jack* merged into spec-vP of the *expect* clause, checking its =d feature (and so picking up the AGENT theta role of *expect*) before moving to spec-TP of this same clause to check case. There are thus no crossing or nested A-movement-1 dependencies in this derivation.

Now consider the alternative derivation for this sentence given in fig 4.4, which yields the incorrect dependencies for this same sentence indicated in 137 above. This time around, *Mary* is again Externally Merged into spec-vP of the *help* clause, but now her D feature is deleted rather than being allowed to persist, immediately revealing her -case feature. When the [trans] little v head of the *try* clause is merged into the structure, therefore, the only option available is to Externally Merge *Jack* into spec-vP, where he incorrectly picks up the AGENT theta role of *try* with his D feature persisting. At this point there are two active A-movement-1 features in the derivation,

<sup>9</sup>Note that the intervention effect between case- and theta-driven movement would also apply here were we to follow Chomsky (1981) in assuming that the ECM object is situated in spec-TP of the infinitival clause where it is assigned case ‘exceptionally’ across a clause boundary.

-case on *Mary* and D on *Jack*, but without DSMC the derivation would continue. *Mary* then subsequently raises to check and delete her -case feature against *expect* as before, after which *Jack* raises to spec-vP of *expect* to pick up the AGENT theta role of that verb, before moving to spec-TP of the same clause once again to check case.

This time around we therefore have two crossing A-movement-1 dependencies and the argument relations derived are clearly incorrect. Crucially, at no point in this derivation were there two moving chains with the same licensee feature as their first feature, meaning that classical SMC cannot rule this derivation out. As we saw, however, there was a point at which there were two chains with the same *types* of feature active, namely the two A-movement-1 features D and -case. DSMC is thus needed here to correctly block this derivation.

At this point, the reader may be wondering about examples such as 140 below, involving the subject control verb *promise* whose direct DP object fails to block control by the matrix subject into the lower clause.

(140) Jack<sub>i</sub> promised Mary<sub>j</sub> to try to t<sub>\*j/i</sub> help.

MGbank implements the analysis from Boeckx et al. (2010) to handle such cases. Under this analysis, example 140 is treated as involving a null dative preposition ([dat]) governing the object of *promise*. This [dat] head therefore checks and deletes both the D and -case features of *Mary* before the resulting PP is merged into the main structure. Example 140 is therefore treated as being structurally parallel to 141 below.

(141) Jack<sub>i</sub> vowed to Mary<sub>i</sub> to try to t<sub>\*j/i</sub> help.

In support of this analysis, note that example 142 below with the subject control verb *promise* is very clearly less degraded than either examples 143 or 144 with ECM and object control verbs respectively.

(142) ??Jack promised to Mary to try to help.

(143) \*Jack expected to Mary to try to help

(144) \*Jack persuaded to Mary to try to help

Furthermore, when the noun *promise* takes a nominal complement, that complement must be headed by *to*, rather than, say, *of*.

(145) Jack's promise to Mary was fulfilled.

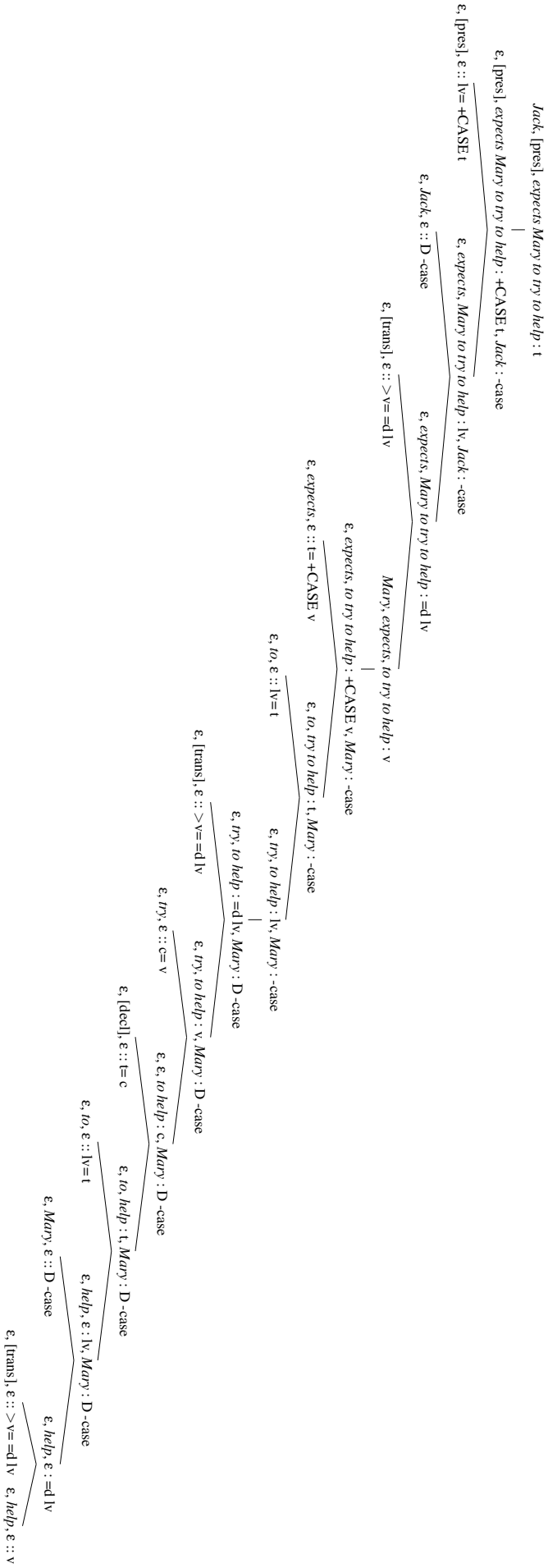


Figure 4.3: An MG derivation tree for the sentence Jack expected Mary to try to help which yields the correct predicate-argument relations.

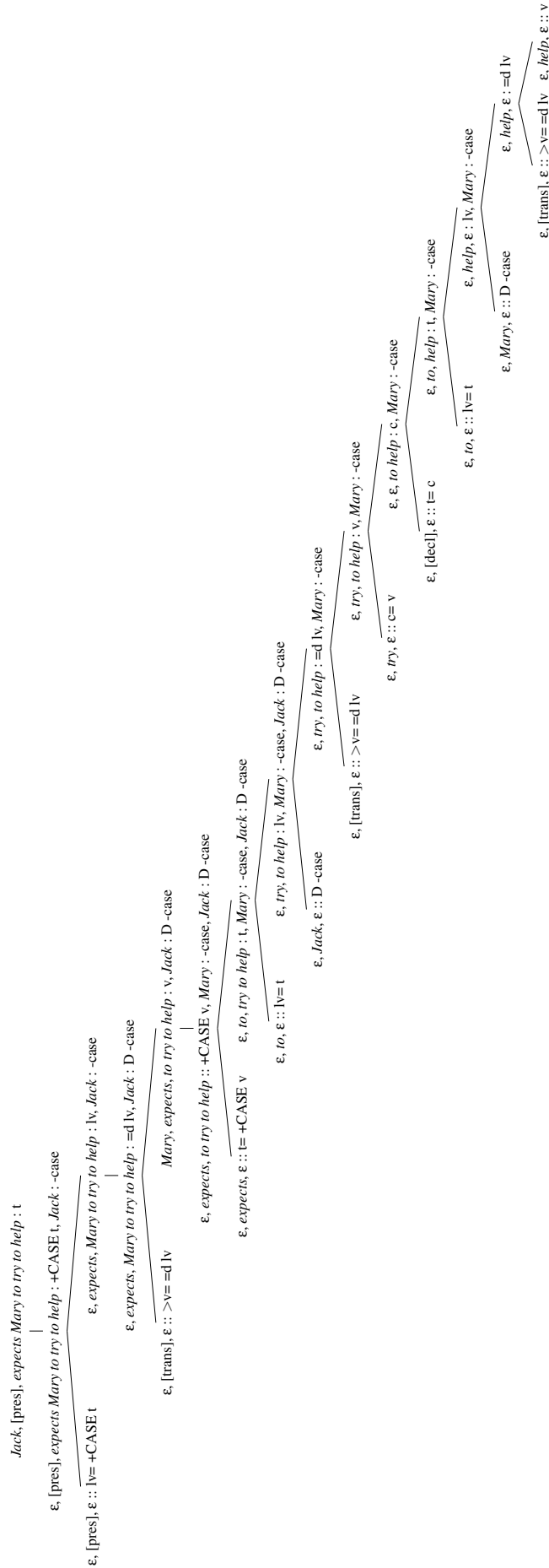


Figure 4.4: An MG derivation tree for the sentence *Jack expected Mary to try to help* which violates DSMC and thus results in *he* incorrectly controlling the null subject of the *try* clause. The derivation is simplified to save space on the page by treating proper nouns as D rather than N, and by omitting the null matrix clause complementizer.

See Boeckx (2010) for further arguments in support of the null preposition analysis of *promise*-type subject control verbs.

In the next two sections we will look at some problems which arose for DSMC in the domain of A-movement, and how these were (partially) overcome.

### 4.2.3 Gerunds and DSMC

One complication for the implementation of DSMC as presented so far comes from examples such as 146 below.

(146) He<sub>i</sub> enjoys [t<sub>i</sub> helping her].

This example features obligatory control into the bracketed gerund clause by *he*. The problem with this is that the gerund itself occurs in an accusative case position as the object of a transitive verb. Adding weight to the argument that gerund clauses have a -case licensee is that they can also appear as the object of a preposition, which standard clauses cannot, as illustrated by the examples below.

(147) He<sub>i</sub> is thinking of [t<sub>i</sub> helping her].

(148) \*He is thinking of (that) he should help her.

The MGBank grammar assumes, following Abney (1987), that gerund clauses of the type seen in 146 consist of an internal verbal heart (projecting up to TP) and an outer DP shell with its own -case licensee.<sup>10</sup> From the perspective of DSMC this represents a problem because this -case feature must be active in the derivation momentarily alongside the D feature of the DP undergoing control movement (in this case *he*). To see this, consider the derivation tree for example 146 given in fig 4.5.

After *enjoys* takes the gerund DP *helping her* as its complement, there are two moving chains present in the derivation, which are: *helping her* : -case, and *he* : D -case. Because both -case and D are A-movement-1 features, this will trigger a DSMC violation. Notice, however, that the -case feature of the gerund is checked and deleted by movement in the very next derivational step. In the MGBank formalism, this problem is therefore circumvented by allowing (in fact requiring) a selecting head to simultaneously check a selector and any number of licensors which sequentially follow it against a matching sequence of selectee and licensee features on the dependent.

<sup>10</sup>Note, however, that even in analyses such as Pires (2001), which reject the proposal of a DP layer, the clausal gerund is still generally assumed to carry a case feature in need of checking.

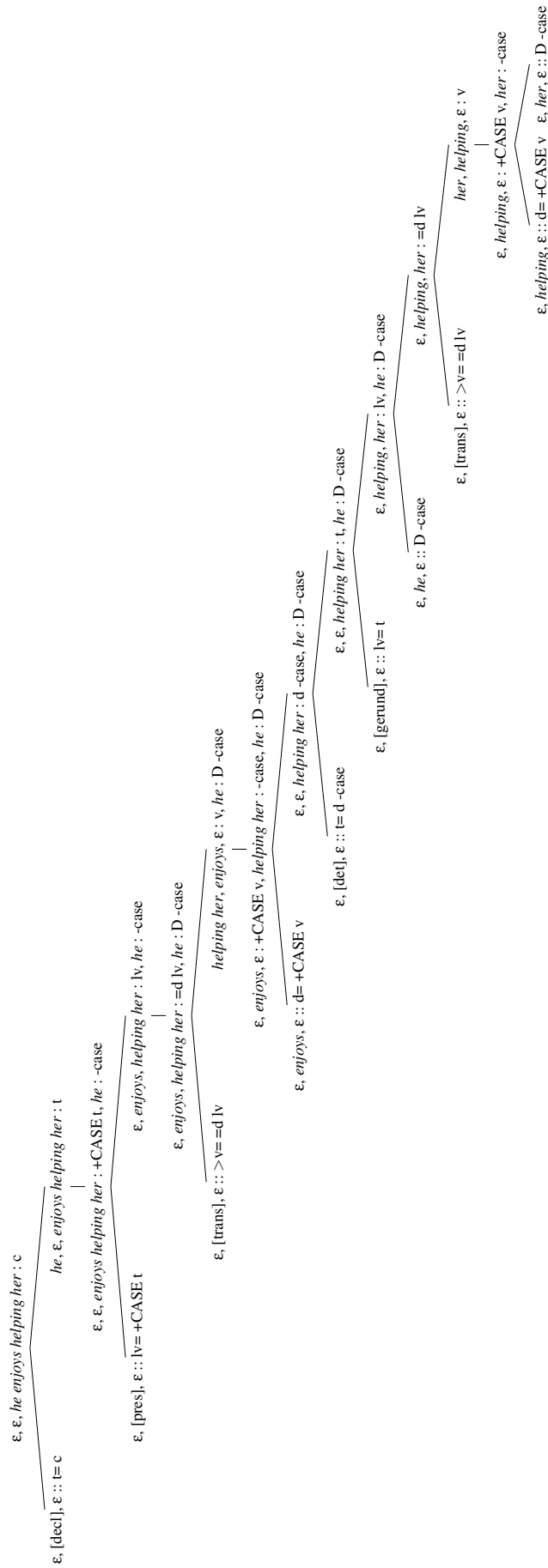


Figure 4.5: An MG derivation tree for the sentence *he enjoys helping her with control into the gerund clause by he.*



Formally, this means composing Merge rules with one or more following Move rules, with the restriction that the dependent whose selectee is checked in the Merge rule must be the one whose licensee(s) are checked in the Move rule(s) that enter into the composition. As a consequence, when the gerund clause is merged as the complement of *enjoys*, it will simultaneously have both its D and -case features checked, thereby correctly bleeding DSMC.

Importantly, classical SMC continues to apply at ALL points in the derivation, including immediately following Merge and between all Move operations.<sup>11</sup> As such, the MCFG-equivalence result of Michaelis (1998) and Harkema (2001) relying on strict enforcement of SMC is unaffected by this relaxation of DSMC.

#### 4.2.4 Reflexive Binding, Principle A and DSMC

This section will look at the MGBank analysis of the binding of reflexive anaphors such as *himself*, *ourselves* etc (the analysis of reciprocal anaphors such as *each other*, *one another* etc is very similar and the relevant lexical categories can be found in Appendix A). We will be specifically concerned here with reflexives which must be locally bound, and will not discuss so-called *picture-noun* reflexives or logophors, which are homophonous with locally bound reflexive anaphors in English but do not require local antecedents (Reinhart and Reuland, 1991, 1993). The discussion will centre on the challenges which were posed by the decision to treat reflexive binding in MGBank as involving a type of A-movement (following Hornstein (2001)) in the context of a grammar constrained by DSMC.

As was noted in section 3.3.6, locally bound reflexive anaphors share certain properties with A-movement traces and OC PRO (which, as discussed in section 3.3.6, we are also treating as an A-movement trace). For instance, all three require local, c-commanding antecedents. In GB, similarities such as these led to all three being treated as anaphors<sup>12</sup> whose distribution was regulated by Principle A of Chomsky's (1981) Binding Theory. This Principle stated that an anaphor must be bound within its *governing category* (also sometimes referred to as its *binding domain*), that category being (roughly) the nearest clause containing a subject. Consider the following examples.

<sup>11</sup>The fact that immediately after the [det] head takes the gerund clause as its complement there are two chains in the derivation with d/D features does not constitute an SMC violation because only one of these chains is moving.

<sup>12</sup>In TG, the term *anaphor* has a much narrower use than elsewhere in linguistics, and is taken to refer exclusively to reflexive and reciprocal anaphors which must be locally bound.

- (149) a. Jack believes himself to be intelligent.  
 b. \*Jack loves herself.  
 c. \*Jack's mother believes himself to be intelligent.  
 d. \*Jack believes himself is intelligent.  
 e. \*Jack believes Mary to love himself.

The unacceptability of 149b arises because *herself* has no suitable antecedent; in 149c, *Jack* fails to c-command *himself*, hence cannot bind it; in 149d *Jack* is not contained in the same tensed clause as *himself*, and so binding again fails; and in 149e, another DP (*Mary*) intervenes and blocks the binding between *Jack* and *himself* (See Hornstein (2001, pages 155-157) for some further interesting interpretive similarities between reflexive binding and control).

Hornstein (2001) argues that rather than unifying A-movement traces, OC PRO and reflexives by treating them all as anaphors regulated by Principle A, we should instead treat them all as residues of A-movement. This, argues Hornstein, allows us to eliminate Principle A of the Binding Theory (and ultimately perhaps even the Binding Theory itself) and capture the locality restrictions of reflexive binding using the same constraints which regulate movement more generally: 149b is then blocked because (simplifying slightly) a reflexive anaphor can be viewed as an overt trace of A-movement, and like all traces it requires a suitable antecedent which is missing in this instance (assuming we exclude the transgender interpretation); 149c is blocked because (remnant movement aside) movement necessarily establishes c-command configurations between the antecedent and its trace, which is not the case here; 149d, meanwhile, is ungrammatical for the same reason that 150 below is, namely that once a DP has its nominative case feature checked by a finite T head it becomes frozen in place for the purposes of A-movement.

- (150) \*Jack<sub>i</sub> believes t<sub>i</sub> is intelligent.

Finally, 149e exhibits a classic intervention effect, parallel to the object control example in 151 below.

- (151) Jack<sub>i</sub> persuaded Mary<sub>j</sub> to t<sub>j/\*i</sub> help.

Despite these and other interpretive similarities discussed by Hornstein, there is one obvious difference between reflexives on the one hand and A-movement traces and OC PRO on the other, which is that (at least in standard cases) only the former are overtly

expressed. This, Hornstein, argues, is a consequence of another important difference between these items: (arguably) only reflexives occur in (accusative) structural case positions. Hornstein suggests that the verb *expect* optionally carries an accusative case licensor feature. If it lacks this licensor, then we see sentences such as 152, with a standard null A-movement trace, if it does not, then we find ECM examples such as 153, with a reflexive.

(152) Jack<sub>i</sub> expects to t<sub>i</sub> be elected.

(153) Jack<sub>i</sub> expects himself<sub>i</sub> to be elected.

Hornstein therefore argues that a reflexive anaphor is an overt residue of A-movement appearing in a cased position. We will now look more closely at the details of Hornstein's analysis, as a version of it is implemented in MGBank. Consider the simple transitive sentence in 154.

(154) Jack<sub>i</sub> likes himself<sub>i</sub>.

Hornstein observes that the two DPs, *Jack* and *himself*, are case marked with different structural cases: *Jack* with nominative and *himself* with accusative. We have seen that case marking generally freezes a DP in place for the purposes of A-movement. Therefore, we cannot regard *himself* simply as an overt trace of *Jack* because the latter would then be unable to move to its surface position. Instead, Hornstein proposes a doubling approach (which is also pursued in Kayne (2002)), under which the derivation starts out by adjoining *self* to *Jack*. Following Chomsky (1995), Hornstein assumes that DPs enter the derivation already specified for case; *Jack* therefore enters the derivation prespecified as nominative, while *self* enters the derivation specified as accusative.

The doubled constituent  $[[Jack] self]$  then merges as the object of *likes*, with the head of this DP, namely *Jack*, picking up the internal theta role of *likes*. *Jack* then subsequently breaks away to move to spec-vP<sup>13</sup> of the *likes* clause (picking up the external theta role), after which it raises to spec-TP to check nominative case. Hornstein assumes that accusative case is checked by covertly raising the DP containing *self* and the trace of *Jack* to the specifier of the verb. Hornstein furthermore proposes that because *self* is necessarily a bound morpheme in English, the pronoun *him* must be

<sup>13</sup>Hornstein does not actually refer to a vP shell layer here, but instead simply assumes that the subject is generated in spec-VP. However, nothing hangs on this, and so I include vP here to make the discussion more consistent with earlier sections of this chapter.

inserted as a grammatical formative to save the derivation from crashing (a Last Resort process that is reminiscent of *do*-support). Thus *self* is realised as *himself*. The basic structure Hornstein proposes for 154 is shown in 155 below, where here strikethrough indicates a deleted copy of a moved constituent (Hornstein adopts the Copy Theory of Movement of Chomsky (1995)<sup>14</sup>).

(155) [<sub>TP</sub> Jack T [<sub>VP</sub> ~~Jack~~ likes [[~~Jack~~]self]]].

Hornstein's approach to reflexive binding and Principle A phenomena is attractive in the present context because, as has been noted already, the only way to establish long-distance dependencies in the MGBank syntax is via movement. We will now look at how an adaptation of this doubling approach can be made to fit into the current formalism, as well as at some problems which arose during its implementation in relation to DSMC and how these were (partially) overcome.

As we have seen, the MGBank grammar does not include a syntactic treatment of morphology. All lexical items enter the derivation as atomic units. Furthermore, the grammar does not include special operations for the insertion of grammatical formatives. Therefore, we must treat *himself* as a unit from the beginning. We can, however, use a null head, which we will label [self] to represent what in a full morphosyntactic account would be instantiated as a separate *self* morpheme. This [self] head will act as the glue which initially combines the antecedent and its reflexive anaphor into a single unit. Assume that this [self] head has the following category (to be revised as the discussion proceeds).

[self] :: d= +case{x} self D -case{x}

We will continue to make the simplifying assumption that proper nouns are simply bare determiners.

Jack :: D -case{ACC.NOM}

While the reflexive *himself* has the following category.

himself :: self= D -case{ACC}

Now consider the derivation tree in fig 4.6 for our example sentence up to the point where the verb selects the doubled *himself+Jack* constituent as its object and checks its -case feature.

<sup>14</sup>See section 3.2.5 for discussion of why the Copy Theory is incompatible with the present strongly derivational framework.

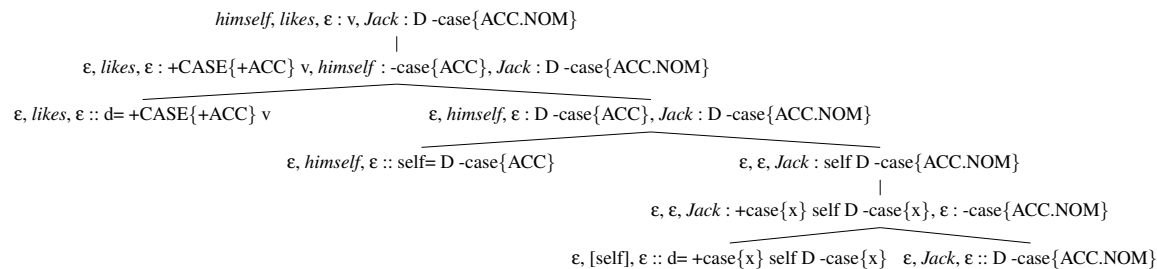


Figure 4.6: A derivation tree for the VP portion of the sentence *Jack likes himself*.

There are several points to note about this derivation. First, there is a new selectee feature, *self*, on the [*self*] head. This feature was used to ensure that the only constituent which can ever select this item is a reflexive (or a reciprocal) anaphor. An alternative approach which would not necessitate expanding the part-of-speech category system in this way would be to use an  $n\{\text{SELF}\}$  selectee instead, and then to have the reflexive select for this selectee by replacing *self=* with  $n\{+\text{SELF}\}=\text{}$ . This seems reasonable as *him* is plausibly a determiner which selects the NP *self* as its complement. At present, MGBank still uses the *self* selectee, however.

A second point of note is that, unlike in Hornstein's approach, the reflexive is not an (adjunct) dependent of *Jack*; rather, *Jack* is selected for by the [*self*] head which in a full morphosyntactic system would be instantiated as a separate overt *self* morpheme. We will return to this point later on.

Finally, notice that in the penultimate stage of this derivation, both D and *-case* co-occur in apparent violation of DSMC. The violation is indeed only apparent, however, because, as we saw in the previous section's discussion of control into gerund clauses, a D *-case* sequence can be checked 'simultaneously' by a *d= +CASE* sequence in the MGBank formalism by composing Merge and Move rules. Therefore, these two features on the doubled DP can be checked simultaneously with DSMC thereby circumvented.

The derivation now continues as shown in fig 4.7, with the null [*trans*] little *v* head selecting the VP and having its *=d* selector checked via control movement of *Jack*, after which the finite T head selects the resulting *vP*, with *Jack* moving to spec-TP to check *-case*.

As in Hornstein's system, in this derivation, *Jack* and the reflexive start out as a doubled DP constituent, with *Jack* then vacating this larger DP and moving to pick up the external theta role of the verb in spec-*vP* and thence to spec-TP to check nomina-

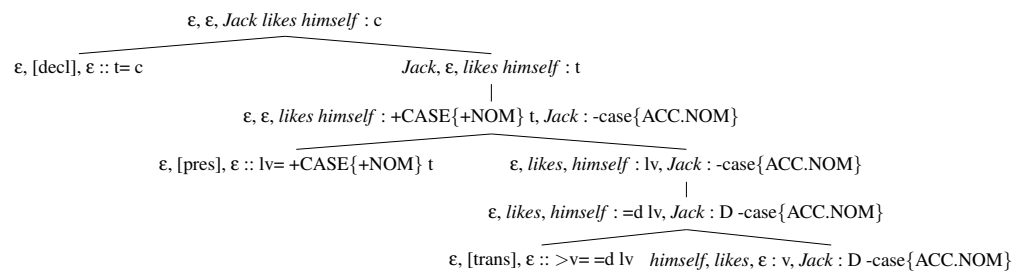


Figure 4.7: An MG derivation tree for the sentence *Jack likes himself*. The derivation of the VP has been abbreviated here but can be seen in full in fig 4.6.

tive case. The remnant DP containing the reflexive and the trace of *Jack* meanwhile also undergoes (remnant) overt movement to check the accusative case of the verb, correctly placing the reflexive in the object position. This operation differs slightly from Hornstein’s approach, however, because we are assuming here that verbs check accusative case overtly, rather than covertly, in English (for reasons to be discussed in section 4.7.3).

Unfortunately, a problem arises for the analysis presented above as soon as we attempt to extend it to cover ECM examples such as 156 below.

(156)  $Jack_i$  expects himself<sub>*i*</sub> to  $t_i$  be amazing.

In MGBank, this sentence would have the following skeletal structure (the prdP analysis is adopted from Mikkelsen (2005) and is in the process of being integrated fully into MGBank, replacing an older analysis in which the subject of the copula was generated in spec-vP, with the copula taking the adjP/DP/PP directly as its complement).

(157)  $[_{TP} Jack_i [_{vP} t_i$  expects<sub>*j*</sub>  $[_{VP}$  [himself  $[t_i$  [self]]]<sub>*k*</sub>  $[_{v'}$   $t_j$   $[_{TP}$  to  $[_{vP}$  be  $[_{prdP}$   $t_k$  amazing]]]]]]].

The prdP is generated using a null [predcatizer] head which selects an adjP as its complement (there are also versions of this head which select for a DP or PP) and then generates the subject as its specifier, with the latter then raising to the surface subject position after *be* (generated in the little v position) takes the prdP as its complement. One problem for the above analysis, whose solution we defer until section 4.3.1.1, is that *Jack* has escaped from a base-generated specifier in violation of the Specifier Island Constraint. A second problem is that the derivation of this structure

will necessarily involve *Jack* and the remnant doubled constituent that contains the reflexive moving alongside one another to check D and -case respectively. Recall that in the derivation of our simpler example 154, there was also a point in the derivation at which -case and D were momentarily simultaneously active. We overcame this problem, however, by allowing the D and -case features of the remnant doubled DP to be checked simultaneously by the verb. This strategy will not help us in the case of 156, however. To see why, assume that we have already derived our doubled DP as before, and that we have also already merged the null prd head with its adjP complement. We now merge the doubled DP into spec-prd, yielding the expression at the root of the abbreviated derivation tree in fig 4.8.

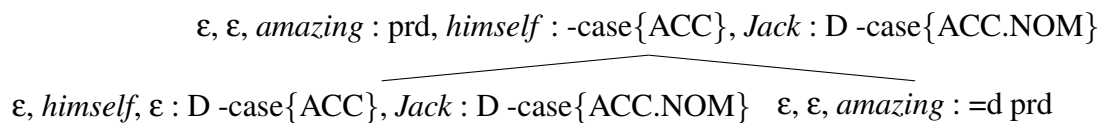


Figure 4.8: Abbreviated derivation tree showing the merger of the doubled constituent *Jack+himself* with the prd' *amazing*. With the type for [self] we have assumed so far, this operation would result in a DSMC violation.

This expression again contains simultaneously active D and -case features, and because the v head has no +CASE licenser, it is not possible to circumvent DSMC by composing Merge and Move rules to delete the D -case sequence on the reflexive simultaneously. What we have here, then, is a genuine violation of DSMC. The problem is created by the fact that one DP moves out of another, and then the two DPs must continue A-moving alongside one another. Note that in the doubling approach of Kayne (2002), the antecedent only vacates the doubled constituent *after* the latter has moved. If implementable, this could have rescued things here. However, in the collapsed tree framework of MGs, the internal phrase structural geometry of expressions is discarded (except that the head chain is identifiable), and constituents must therefore begin moving the moment they have been Externally Merged, if they are ever to move at all. Kayne's perspective will therefore not help us here.

We can approach this problem from a conceptual standpoint by first observing that at a semantic level there is no intervention effect in 156 precisely because *Jack* and *himself* refer to the same individual, hence there is no ambiguity and the sentence is fine. What is needed, therefore, is some way to circumvent DSMC just for the case where the D and -case features ultimately belong to the same referent, while retaining

its constraining effect in all other cases. One could conceivably extend the formalism with mechanisms to achieve this.

MGBank does not include such extensions at present, however. Instead, it includes a -self licensee on the [self] head whose purpose is to shield the D feature of the antecedent for DSMC purposes from the -case feature of the remnant doubled DP containing the reflexive. This is the reason that the definition of DSMC in section 4.2 included two groups of A-movement features. -self appears in the group A-movement-2 and hence does not conflict with -case (or D) which appears in the group A-movement-1.

The -self licensee is only introduced into the derivation by a reflexive (or reciprocal) anaphor, and is always checked and deleted before any other active DPs end up in the structure, meaning that it never circumvents DSMC except where it should. This is achieved by placing the +self! licenser that deletes this feature in front of the =d on the [trans] little v head, meaning that its -self is deleted before the external argument is introduced. And any internal DP arguments which are introduced into the structure will of course have their D and -case features deleted simultaneously by the verbs or prepositions governing them, meaning that they too are never active alongside -self. The ! on the +self! licenser marks this feature as a deleting suicidal licenser (see section 3.3.4), ensuring that where no reflexive is present, the feature will simply self-delete.

Our new categories for the [self] morpheme and [trans] head are thus as follows.

$$\begin{aligned} [\text{self}] &:: d= +\text{case}\{x\} \text{ self } -\text{self } D -\text{case}\{x\} \\ [\text{trans}] &:: >v= +\text{self! } =d \text{ lv} \end{aligned}$$

Assuming all other categories to be the same as before, the stage at which the doubled constituent is merged with the *prd'* *amazing* will now look as in fig 4.9.

There is now no DSMC at this stage of the derivation, because -case and D are no long simultaneously active. The derivation subsequently proceeds as shown in fig 4.10.

Next, consider the following example.

(158) Jack<sub>i</sub> showed Mary himself<sub>i</sub> (in the mirror).

Of interest here is the fact that the object *Mary* fails to intervene between the binder *Jack* and the reflexive. For this reason, Chomsky (1981) formulated Principle A of his



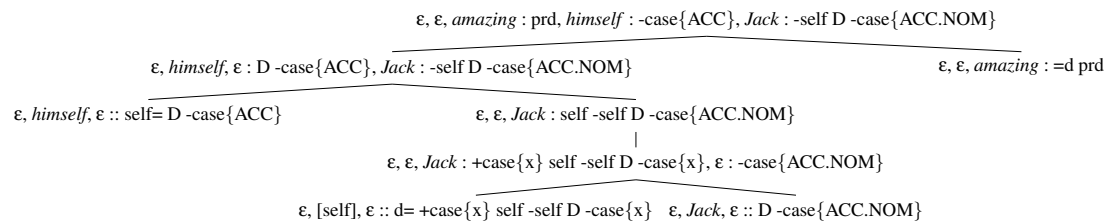


Figure 4.9: Abbreviated derivation tree showing the merger of the doubled constituent *Jack+himself* with the *v'* *be amazing*. This time there is no DSMC effect owing to the addition of the *-self* licensee on [*self*].

binding theory (which was responsible for regulating the distribution of reflexive and reciprocal anaphors) in the following terms.

- (159) Principle A: An anaphor must be bound in its governing category.
- (160)  $\beta$  is a governing category for  $\alpha$  if and only if  $\beta$  is the minimal category containing  $\alpha$ , a governor of  $\alpha$ , and a SUBJECT accessible to  $\alpha$ .

In this definition, the term SUBJECT is a generalisation of the term *subject*. The technicalities of Chomsky's definition need not concern us here. The important point to note is that as one might expect a SUBJECT includes a traditional subject such as *Jack* in 172 but excludes the object of a simple transitive verb, in this case *Mary*. The governing category of *himself* is therefore the TP containing the subject *Jack* meaning that the latter is able to bind the reflexive in spite of the intervening object. This contrasts with the case in 149e, repeated as 161 below, in which *Mary* (or a trace of *Mary* in a raising-to-object analysis of ECMs) counts as the SUBJECT of the infinitival clause. Since this clause excludes *Jack*, the latter is not contained within the governing category of the anaphor and hence is unable to bind it.

- (161) \**Jack* believes *Mary* to love *himself*.

In our A-movement-based approach to reflexive binding, which makes no reference to governing categories/binding domains, it may at first appear as though intervention effects in the form of DSMC will incorrectly cause the object in 172 to intervene. In the system of Chomsky (1995) we could appeal at this point to the fact that both objects are generated in the same Minimal Domain and hence are Equidistant from the subject. In MGs there is no notion of domain whatsoever, however. Nevertheless, example 172 is unproblematic here, even if we assume, as MGBank does, that *Mary* is a bare DP

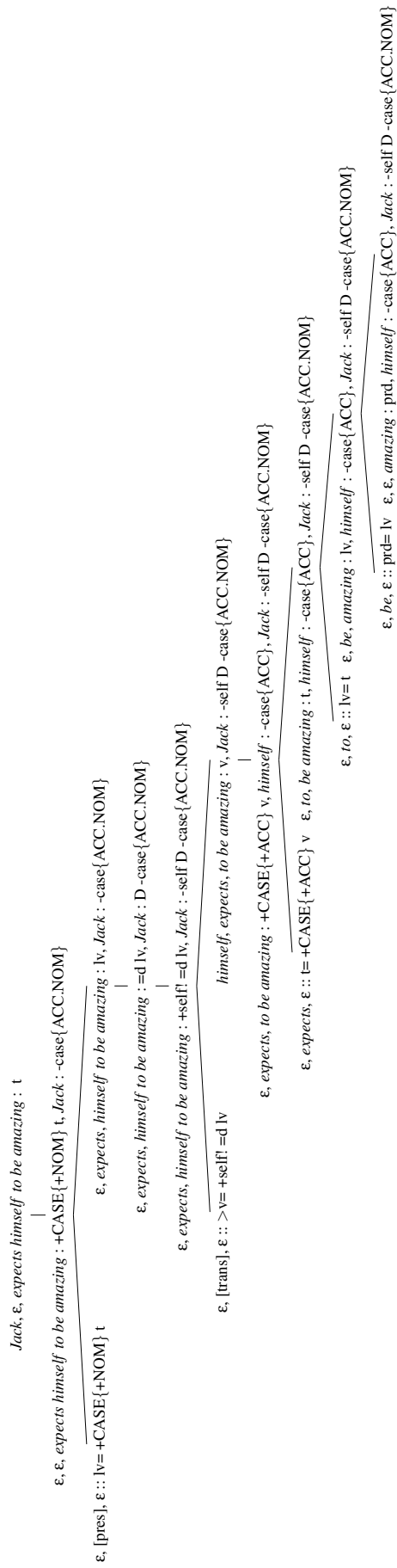


Figure 4.10: Abbreviated derivation tree for the sentence *Jack expected himself to be amazing* up to TP. See also fig 4.9 for the derivation of the *prdP*.

and not one which is governed by a null case-checking [dat] prepositional head. To see why, we must look at the MGbank type for a double object verb such as *show*, which is as follows.

show :: d= +CASE{+ACC} =d +CASE{+ACC} v

Because the =d +CASE sequence which checks the leftmost object's A-features is sequential, *Mary's* D and -case features can be checked and deleted 'simultaneously' by composing the Merge and Move involved in checking them, as was proposed in section 4.2.3. As a result, they are knocked out before DSMC has a chance to apply and the derivation proceeds without issue.

At this point, the reader may be wondering whether *Mary* will correctly intervene in the ECM object example in 161. In fact *Mary* does correctly intervene in this instance because, unlike in 172 where a single V head checks both of *Mary's* A-features, here these features are checked by two *different* heads (D by the lower [trans] head and -case by *believes*). This means that it is not possible for these features to be checked simultaneously. Furthermore, the -self feature on *Jack* will have already been eliminated by the [trans] head of the *love* clause before *Mary* is introduced. With -self deleted, *Jack's* D feature will be exposed and be active alongside *Mary's* D feature, thereby triggering a DSMC violation and correctly blocking the derivation.

What about the superficially similar object control example below, which also exhibits this intervention effect?

(162) \**Jack<sub>i</sub>* persuaded *Mary* to love himself<sub>*i*</sub>.

Here *persuade* not only checks the case of its object DP, but also first checks its D feature, thereby assigning it a theta role. As we saw in section 3.2.7, in MGbank, ECM and object control verbs have the following distinct syntactic types.<sup>15</sup>

believe :: t= +CASE v  
persuade :: c= =d +CASE v

Given that *persuade*, like the double object verb *show*, has a =d +CASE sequence, in principle the simultaneous feature checking strategy should be available. However, as we will see in section 4.7.4, for quite independent reasons having to do with the need to constrain the distribution of non-obligatorily controlled PRO, the MGbank

<sup>15</sup>Note that the fact that ECM verbs select a TP clausal argument whereas object control verbs select a full CP is an artefact from mainstream Minimalist analyses which assume CP, but not TP, to be a phase barrier.

grammar enforces a locally determined Move-over-Merge constraint.<sup>16</sup> According to this constraint, all Move rules which can apply following a Merge operation must apply before Merge can apply again. At the point when *persuaded* is looking to assign an internal theta role to a DP, therefore, *Jack*, having already picked up the internal and external theta roles of *love*, and having had his -self licensee deleted by [trans] of the lower clause, will now be available to move and check *persuade*'s =d feature. Given Move-over-Merge, *Mary* cannot be Externally Merged to pick up this role because *Jack* is able to check it via Move.

Note that the Move-over-Merge constraint is strictly locally determined, and hence involves no look ahead. The fact that the derivation is doomed to crash from this point on (because *Jack* cannot both check *persuade*'s +CASE licenser and then go on to pick up the latter's external theta role) is therefore irrelevant. Note also that we cannot save things by deleting *Jack*'s D feature after he picks up the external theta role of *love* (allowing *Mary* to be merged into the structure with simultaneous feature checking) because *Jack*'s D feature needs to persist to allow him to later pick up the external theta role of *persuade*.

What about the other locality effects which were observed to hold for reflexive binding, just as they do for A-movement and obligatory control? In our system, a reflexive will necessarily have an antecedent because it always starts the derivation in a doubled constituent containing that antecedent. This is not actually sufficient to block 149b, repeated below as 163, however, because there is at present nothing in our grammar to prevent *Mary* from being the antecedent of *himself*, despite the lack of gender agreement.

(163) \**Mary<sub>i</sub> loves himself<sub>i</sub>*.

There is also currently nothing enforcing person and number agreement between the reflexive and its antecedent, meaning that our grammar also overgenerates 164 and 165.

(164) \**Mary<sub>i</sub> loves myself<sub>i</sub>*.

(165) \**Mary<sub>i</sub> loves themselves<sub>i</sub>*.

We can use the fine-grained selectional feature system to block such examples. For example, we can rule out 163 by updating our lexicon as follows.

<sup>16</sup>This is, somewhat ironically, the precise opposite of Chomsky's (2000) Merge-over-Move constraint.

[self] :: d{x}= +case{y} self{x} -self D -case{y}  
 herself :: self{+3SG.+FEM}= D -case{ACC}  
 Jack :: D{3SG.MASC} -case{3SG.ACC.NOM}

When [self] selects the antecedent, the latter's gender and agreement properties will percolate onto the self selectee of the resulting constituent owing to the  $x$  variables on [self]; these gender and agreement properties can then be selected for by the reflexive, whose +3SG, +FEM requirements will ensure person, number and gender agreement between antecedent and anaphor.

The c-command requirement between the antecedent and anaphor also holds in MGbank because, as we have seen, movement in the present formalism necessarily establishes c-command configurations (remnant movement aside). Thus, 149c, repeated as 166, is correctly underivable.

(166) \*Jack<sub>*i*</sub>'s mother believes himself<sub>*i*</sub> to t<sub>*i*</sub> be intelligent.

Note, furthermore, that it will never be the case that the reflexive ends up c-commanding its antecedent, because it is the antecedent which bears the -self licensee which shields its D and -case features. As soon as -self is deleted, therefore, the D feature of the antecedent will be revealed and will trigger a (D)SMC violation if the reflexive is still active in the derivation. This means that 167 is also correctly blocked.

(167) \*Mary believes himself<sub>*i*</sub> to love Jack<sub>*i*</sub>.

What about the fact that a finite clause barrier also blocks binding, as in 149d above and the similar 168 below?

(168) \*Jack<sub>*i*</sub> expects [<sub>*CP*<sub>*fin*</sub></sub> himself<sub>*i*</sub> is intelligent].

Reflexive/reciprocal anaphor binding aside, in MGbank, the finite-clause boundedness of case- and theta-driven A-movement is a consequence of the fact that the last feature in a sequence of A-movement licensees is a DP's -case feature. Like all other licensees (with the exception of the theta licensee D), -case cannot persist after being checked. Finite T (unlike infinitival T) has a +CASE licenser which must be checked as soon as finite T is merged with its vP complement. Given DSMC, there can only be one DP moving to check case or theta features in the derivation at this point. And this DP must check its -case feature against T, which will freeze it in place as far as A-movement is concerned (though it can then undergo A'-movements, of course).

However, in the case of 168, we could indeed have the two DPs, *himself* and *Jack*, in the derivation at the moment when the embedded finite T is introduced into the structure, since these will have been generated in spec-vP initially as a single doubled constituent. Furthermore, the +self! licenser (which appears on copula *be* as well as [trans]) will have already self-deleted by the time this doubled constituent is introduced. The -self shield on *Jack* will therefore remain intact, allowing both DPs to harmoniously coexist. The reflexive could then in principle check the nominative case feature of this finite T by moving to spec-TP, after which *Jack* could escape into the upstairs clause, deleting -self and D features when the upstairs [trans] head is introduced, and eventually moving to the matrix spec-TP to check nominative case.

As was noted above, Hornstein argues that the ungrammaticality of 168 follows from the fact that nominative case positions freeze a DP in place, hence *Jack* cannot move from spec-TP of the embedded finite clause, just as he cannot in 169 below.

(169) \* $Jack_i$  expects [ $TP_{fin}$   $t_i$  is intelligent].

Crucially, Hornstein assumes with Chomsky (1995) that DPs enter the derivation already specified for case (i.e. as nominative, accusative or genitive), rather than receiving a case assignment during the course of the derivation. *Jack* therefore enters the derivation already specified as nominative, while the reflexive is specified for accusative case. This means that the case licenser on finite T can only be checked by *Jack* which as a consequence will correctly be frozen in place, blocking the derivation of 168. On the other hand, 170 below is grammatical because infinitival T is not a case assigner (or phi-feature checker) in English, hence *Jack* is able to escape the embedded clause here.

(170)  $Jack_i$  expects [ $TP_{inf}$  to  $t_i$  be intelligent].

Hornstein's strategy for blocking reflexive subjects of finite clauses was in fact already implemented here by the exclusion of a NOM selectional property on the -case feature of reflexive pronouns such as *himself*; these items are marked only with ACC, meaning that in 168, the finite T will reject *himself* as a checker of its +CASE{+NOM} licenser. Proper names such as *Jack* (or rather, the null determiners which govern these nouns in the actual MGBank grammar), meanwhile, are syncretised for all cases by being marked with NOM, ACC and GEN, meaning that *Jack* is able check finite T's +CASE{+NOM} licenser. However, it will then be frozen place meaning, that it is impossible to derive 168.

Returning to the binding of reflexives in double object constructions, in addition to 172 above, in which the subject binds the rightmost object, we also find 171, with the subject binding the leftmost object, and 172 with the leftmost object binding the rightmost object. This is expected under Principle A because in both cases the reflexive is bound within its governing category, i.e. the finite TP containing the subject.

(171) Jack<sub>i</sub> showed himself<sub>i</sub> Mary.

(172) Jack showed Mary<sub>i</sub> herself<sub>i</sub>.

Example 171 is unproblematic for the current approach: the rightmost object is first selected by *show* and has its D and -case features checked, after which the verb selects as its leftmost object the doubled constituent containing the reflexive and its antecedent, checking the reflexive's D and -case features; the antecedent *Jack* then raises to spec-vP to check -self and D and then raises further to spec-TP to check -case.

However, our grammar currently undergenerates 172. To see this, assume that we have reached the stage where the doubled constituent has been constructed and the verb has just selected this item as its complement and checked its -case licensee. This stage is shown in fig 4.11.

The problem here is that the -self shield is now blocking the D feature on *Mary* from being checked by *show* via control movement. This is because we have not yet reached the stage in the derivation at which the [trans] little v head deletes -self. The cause of this issue was that earlier on in the derivation, when [self] selected *Mary*, it effectively pied-piped the D and -case features of *Mary* by deleting them and replacing them with its own which are tucked behind its -self licensee. As a consequence, *Mary* is effectively transformed from a DP with the feature sequence: D -case, into one with the sequence: -self D -case, in which the D feature is shielded, preventing us from generating 172. What we need then, is some way to make this 'pied-piping' optional (since we do need it in other examples as we have seen). We can do this by introducing the following alternative version of the [self] head into the grammar alongside the one we have been using so far.

$$[\text{self}] :: d\{x\} = \text{self}\{x\} \text{-self}$$

This head is the same as the other one, except that it lacks +case and D features. This means that when it selects *Mary*, it will not 'pied-pipe' the latter's D and -case features. Instead, the D feature on *Mary* can be allowed to persist and subsequently be

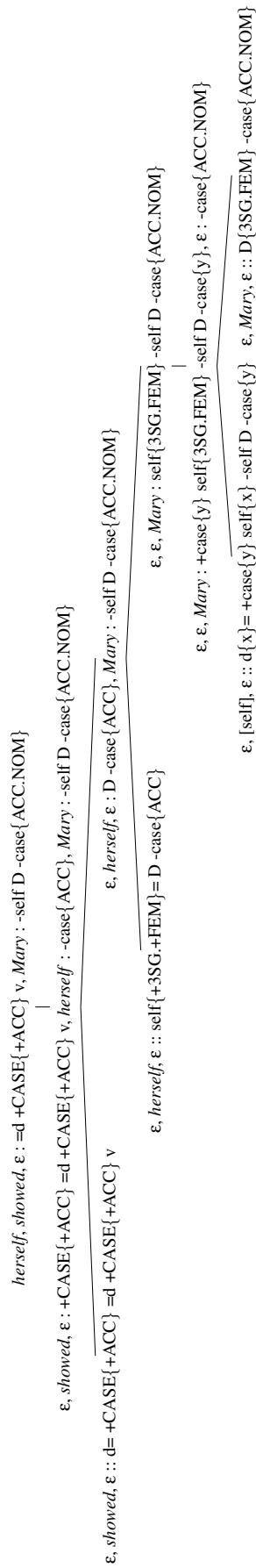


Figure 4.11: Derivation tree showing the Merger of the doubled antecedent-reflexive DP into the complement position of *showed*.



used to check and delete the =d on *showed* via control movement, after which *Mary* will move to the outermost spec-VP position to check -case. This time around, then, *Mary* is kept in a separate chain from [self], which after being selected by the reflexive will only have a single -self feature to check. This will later be accomplished by moving the null [self] chain to spec-vP. Note that this movement will necessarily be covert, because [self] has no phonetic content (since it did not attach to *Mary*).

The derivation of the v' *showed Mary herself* up to point where +self! and -self are checked and deleted is shown in fig 4.12. The derivational steps which follow this should be clear at this stage in our discussion.

This same non-pied-piping [self] morpheme can also be used to generate examples such as 173, where the reflexive is governed by a preposition.

(173) Jack showed *Mary<sub>i</sub>* to *herself<sub>i</sub>*.

On the other hand, the MGBank grammar correctly blocks the derivation of examples such as 174 and 175 below.

(174) \*Jack talked to *herself<sub>i</sub>* about *Mary<sub>i</sub>*.

(175) \*Jack showed *herself<sub>i</sub>* *Mary<sub>i</sub>*.

Example 174 is blocked because *herself* is governed by a preposition, hence it does not c-command *Mary* meaning that there can be no movement dependency between them. Example 175 is ruled out because the reflexive's -case feature must always be checked before that of the antecedent DP (since the antecedent DP's -self shield cannot be deleted until the -case feature of the reflexive has already been eliminated or a DSMC violation will be triggered). This means that the second +CASE licenser on *showed* must be checked by the antecedent which will place it to the left of the reflexive.

Unfortunately, for the same reason that the grammar correctly blocks 175, it also incorrectly blocks 176 below, owing to the fact that the antecedent of the reflexive is governed by a preposition. As a result, no c-command configuration between antecedent and anaphor holds, indicating that movement to the antecedent position from the anaphoric position is impossible.<sup>17</sup> MGBank therefore currently undergenerates examples such as this.

<sup>17</sup>As we have seen, in the current formalism, movement always establishes c-command configurations (remnant movement aside) and is the only means by which long-distance dependencies can be formed. As such, the c-command requirement on intervention effects is absolute here. In the MP literature it is sometimes argued that there is a type of defective intervention in which a DP which has already had its A-features checked and deleted (say by a preposition) can still act as an intervener. For example,

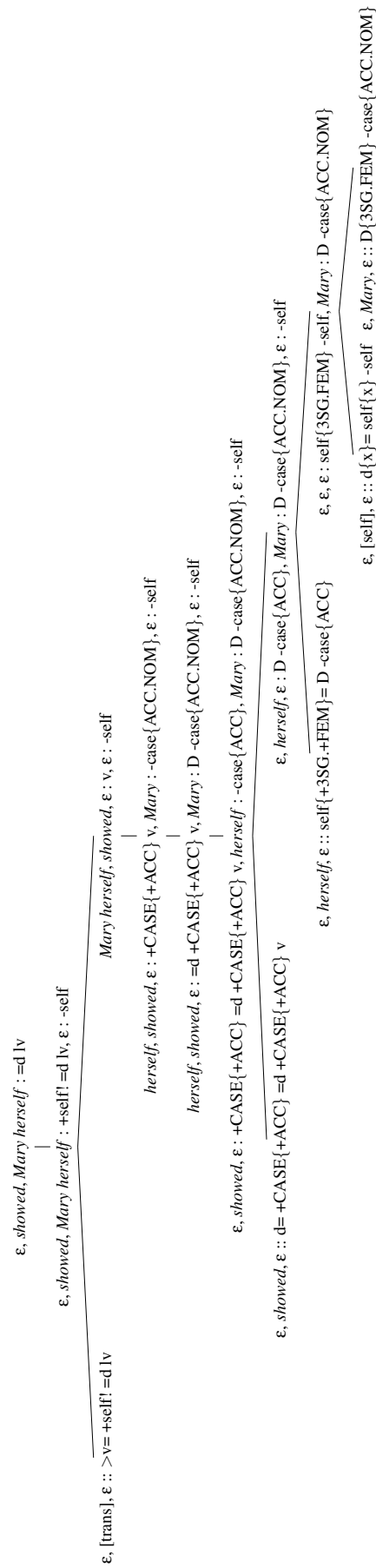


Figure 4.12: Derivation tree for the *v'* showed *Mary herself* in a sentence like *Jack showed Mary herself (in the mirror)*.

(176) Jack talked to Mary<sub>i</sub> about herself<sub>i</sub>.

It is worth noting two points of divergence between the approach adopted here and that proposed in Hornstein (2001). The first is that whereas in Hornstein's system the nominal *self* morpheme adjoins to the antecedent DP, here the [self] head (which in a full morphosyntactic system would be replaced by a bound morpheme *self*), selects the antecedent DP as its complement. The reason for this is of course that for many constructions we require the antecedent to end up associated with a -self licensee to shield its D feature and avoid DSMC violations. This cannot happen if [self] (or himself+[self]) adjoins to the antecedent DP, since in an adjunction structure it is the features of the selectee, not the selector, which end up on the head of the resulting constituent. Note also that placing -self on the antecedent rather than on the reflexive, and then having this licensee checked by [trans], is necessary for ensuring that the antecedent always c-commands the anaphor and not vice versa. Hornstein (2001, pages 163-164) does present some evidence from wh-island violations that *self* is indeed an adjunct, but I could not work out an analysis in which this was true and which could also cover the same range of positive and negative data as the one presented here.

The second point of divergence is that Hornstein assumes that while *self* can bear case, it cannot bear a theta role. In effect, it exists solely to allow the antecedent DP to check two case features (on pages 164-165 Hornstein does discuss one possible semantic function of *self*, but this does not involve it bearing a theta role). However, in the MGBank grammar, it is sometimes necessary to allow the reflexive to also check theta roles. Consider the following example.

(177) Jack<sub>i</sub> believed himself<sub>i</sub> to have been persuaded t<sub>i</sub> to t<sub>i</sub> help.

---

consider the following contrast, presented in Hartman (2011) (traces, PRO and indices have been added).

- (1) a. It is important (to Mary) [<sub>CP</sub> to PRO avoid cholesterol].  
 b. Cholesterol<sub>i</sub> is important (\*to Mary) [<sub>CP</sub> t<sub>i</sub> to PRO avoid t<sub>i</sub>].

1a with an expletive subject is perfectly acceptable, with or without the dative experiencer. In the tough-movement construction in 1b, *cholesterol* (after checking case within the verb phrase) undergoes wh-movement to spec-CP of the embedded clause before undergoing A-movement to the subject position of the matrix clause (tough movement is therefore treated as an instance of so-called *improper movement* both in Hartman's system and in the MGBank grammar - see section 6.4.2.1). In this case, the presence of the dative experiencer leads to a deviant sentence, which Hartman argues to be a case of defective intervention of the final A-movement by the dative experiencer, which has already had its A-features checked and deleted. The MGBank grammar is not capable of replicating this intervention effect because neither SMC nor DSMC can apply here in the absence of any active A-features on the experiencer at the point at which the PP is merged into the main structure. See Bruening (2014) for arguments against the existence of such defective intervention.

In this example, *persuaded* is a passivised object control verb. As we saw above, unlike ECM verbs like *believe*, object control verbs not only check the case of their DP objects, but they also assign them a theta role, which in the current context means they check their D feature. Passivization is standardly assumed to eliminate a verb's case assigning ability (along with its ability to assign an external (but not an internal) theta role - *Burzio's Generalisation*). Therefore, a passivised object control verb like *persuaded* in MGBank has the following category, with the +CASE licenser that follows the =d on the active version being absent.

persuaded :: c= =d v

Let's assume that we have already constructed the infinitival CP *to help*, which contains the moving reflexive and its antecedent, the doubling constituent having been generated in spec-vP of the [trans] head. Fig 4.13 shows the stage of the derivation at which this CP is merged with *persuaded*.

$$\begin{array}{l} \varepsilon, \textit{persuaded}, \textit{to help} : =d v, \textit{himself} : -\textit{case}\{\textit{ACC}\}, \textit{Jack} : -\textit{self D} -\textit{case}\{\textit{ACC.NOM}\} \\ \varepsilon, \textit{persuaded}, \varepsilon :: c= =d v \quad \varepsilon, \varepsilon, \textit{to help} : c, \textit{himself} : -\textit{case}\{\textit{ACC}\}, \textit{Jack} : -\textit{self D} -\textit{case}\{\textit{ACC.NOM}\} \end{array}$$

Figure 4.13: An abbreviated derivation tree for the *V'* *persuaded to help* in which the reflexive's D feature has not persisted. This derivation is doomed because there is no way to check the =d selector of *persuaded*.

The problem here is that there is no D feature available to check the =d of *persuaded*: the D feature of *himself* has already been eliminated (when the [trans] head of the lower *help* clause selected the doubled constituent as its specifier), and the D feature of *Jack* is current (and necessarily given the -case feature on the reflexive) shielded by the -self licensee. Nor can we Externally Merge another DP to check the =d selector owing to the fact that *persuaded*, being passive, has no +CASE feature to allow it to simultaneously check the -case licensee of this incoming DP; this means that an SMC violation would be triggered owing to the fact that the incoming DP and the reflexive would both have active -case licensees in the derivation at the same time.

We can overcome this impasse, however, if instead of the D feature on the reflexive being deleted after it is checked by the lower [trans], it is permitted to persist. In that case, the reflexive can subsequently check and delete its D feature against the =d feature of *persuaded* before moving on to check and delete its -case feature against the

+CASE licenser of the ECM verb *believe*. At this point, the [trans] head that takes the VP headed by *believe* will be introduced into the structure, checking the -self and D features of *Jack* before the latter finally moves to spec-TP of that clause to check its -case feature (note that the -self feature on *Jack* will not be deleted by the [intrans] head of the *persuaded* clause, because the grammar includes a version of [intrans] which lacks the +self! licenser<sup>18</sup>). The derivation tree for this sentence is left as an exercise to the reader.

Some further instances where the MGBank currently undergenerates should be noted before closing this section. One case is illustrated by 178 below.

(178)  $Jack_i$  believed himself<sub>*i*</sub> to have been persuaded  $t_i$  to want to  $t_i$  help.

This example is problematic because the introduction of the transitive *want* clause above the *help* clause means that the -self shield on *Jack* will be eliminated too early by the [trans] head of the *want* clause. This will either trigger an SMC or a DSMC violation depending on whether the reflexive's active feature is D or -case. One way around this would be to allow -SELF to persist in the same manner that D does. This would mean that a -SELF on *Jack* could survive being checked by the +SELF! of the *want* clause and only later be deleted by the +SELF! of the *believe* clause. Unfortunately, adopting this strategy means that we immediately overgenerate examples such as 161 above, repeated as 180 below.

(179) \* $Jack_i$  believes Mary to love himself<sub>*i*</sub>.

The reason is that *Jack* now need not have his -SELF feature deleted by the [trans] head of the *love* clause. This allows *Jack* to pass by *Mary* unproblematically, and to check and delete -SELF against the [trans] head of the *believes* clause. Ultimately, a better approach to reflexive/reciprocal binding may be to enrich the formalism with some mechanism for ensuring that an antecedent and its overt anaphor do not trigger DSMC violations, which would allow us to eliminate the -self licensee. This is left for future research.

---

<sup>18</sup>The reason for including another version of [intrans] little *v* which *does* have a +self licenser is that intransitive verbs can be modified by adjuncts containing reflexives, as in *he arrived by himself*. The MGBank grammar is forced to treat the PPs in such examples as complements, rather than adjuncts, however, since otherwise there is no way to move *he* out of the adjunct PP to pick up the internal theta role of *arrived* (across-the-board movement does not work in this case owing to an SMC violation which occurs inside the adjunct PP; the interested reader is invited to work through the derivation to see at which point this occurs.)

Finally, examples such as 180 below are also currently not generated by the grammar, owing to the fact that two active -self features would be required to co-exist in the derivation simultaneously.

(180) \* $Jack_i$  believes  $himself_i$  to love  $himself_i$ .

#### 4.2.4.1 Across-the-board movement and binding

The MGbank grammar is able to generate some examples involving multiple co-indexed anaphors, however, such as 181 below.

(181)  $Jack_i$  showed  $himself_i$  to  $himself_i$  (in the mirror).

(182)  $Jack_i$  showed  $himself_i$   $himself_i$  (in the mirror).

The challenge of these examples (which are to my ears both perfectly acceptable) is to explain how there can be two reflexive pronouns in each but only one DP antecedent given our doubling approach to reflexive binding. The answer is that we can use the across-the-board phrasal movement mechanism which was introduced in section 3.3.7.2 for coordination, adjunct control and several other phenomena, in order to unify what will initially be two instances of the antecedent *Jack*, one inside the rightmost complement and one inside the leftmost one. The precise derivations are left to the reader as an exercise (although these sentences are also included in MGbank and can therefore be inspected using the Autobank system introduced in the next chapter). Note that this same mechanism also allows us to generate structures which are equivalent to those seen in adjunct control, featuring the binding of a reflexive sat inside an adjunct clause.

(183)  $Jack_i$  helped  $Pete_j$  while also helping  $himself_{i/*j}$ .

Here, *himself* can only be bound by *Jack*, not by *Pete*. In MGbank this effect holds as a simple consequence of the fact that the first available adjunction site for the adjunct clause is the VP, and by this stage *Pete* has already had both his D and -case features checked, meaning that across-the-board movement of a second instance of *Pete* out from the adjunct clause island is impossible. Adjoining the adjunct clause to vP, however, means that there will be versions of *Jack* moving for -case inside both the adjunct and the main structure. The fact that one started out in a doubled constituent with the reflexive while the other did not is irrelevant here, because the ATB checks

do not look at already deleted features;<sup>19</sup> all that matters, therefore, is that the two chains have identical unchecked feature sequences at this point in the derivation. The two instances of *Jack* can therefore be unified, with the single instance then moving to check -case in spec-TP.

Finally, the ATB mechanism can also be used to generate the following example, which (Chomsky, 1981, pages 44-55) argued to constitute evidence of successive cyclic A-movement via spec-*to* (so that the reciprocal anaphor could be bound by the trace of *they* in the *appear* clause, though arguably this was not actually required by his definition of Principle A anyway, as noted in (Boeckx, 2009, page 30)).

(184) they<sub>i</sub> [<sub>VP</sub> t<sub>i</sub> are likely [<sub>TP</sub> to appear [<sub>TP</sub> to [each other]<sub>i</sub> [<sub>TP</sub> to t<sub>i</sub> be happy]]]].

All that is needed here is to include a +self! licenser after the +case licenser on the preposition *to* so that -self is eliminated on the instance of *they* which starts out by being combined with the reciprocal anaphor inside the PP. With this feature eliminated, both this instance of *they* and the instance in the main structure that starts out as the subject of *be happy* will have the same D -case sequence, allowing them to be unified when the PP *to each other* is merged into the main structure. As indicated by the lack of intermediate traces, this analysis does not require successive cyclic movement via spec-*to*, in accordance with the framework of Epstein and Seely (2006) which we adopt here (see section 4.7.3).

#### 4.2.5 Covert-only and overt-only licensees

We have seen that whether movement is overt or covert is determined by the licenser features and that as a result the parser must initially pursue both options. However, there are certain types of movement which are always either overt or covert, at least in particular languages. Movement for control, for example, is exclusively overt in English,<sup>20</sup> while movement for polarity item licensing is exclusively covert. In order

<sup>19</sup>In this sense the approach taken to ATB movement here differs from that in Kobele (2008), where two unified chains must have the same derivational pasts, in the sense that as well as sharing the same sequence of checked features, they must also have the same sequence of already checked and deleted ones.

<sup>20</sup>Note, however, that one advantage a movement account of control arguably has over both PRO-based approaches and approaches which treat control as a lexicalised property of a verb, is that it can straightforwardly handle the cases of obligatory backward control which are apparently found in certain languages, and which are discussed at length in Boeckx et al. (2010). In backward control constructions, it is the controlled element which is overtly expressed while the controller is the null argument. An example of a language with backward subject control is Tsez. Consider the following two examples from Polinsky and Potsdam (2002) (cited in Boeckx et al. (2010, pages 106-107)).

to reduce the search space of the parser and improve its efficiency, certain licensees are therefore specified as only being able to trigger either overt or covert movement, enabling the parser to immediately disregard the other option. The overt-only licensees are D, -*ep*, -*foc*, -*top*, -*n* and -*tough*, while the covert-only movers are -*pol* and -*neg*s.

Restricting certain licensees to being covert only is a particularly welcome move from the perspective of the theoretical time complexity of MG parsing because the number of possible instantiations of a particular covertly moving chain is not dependent on the input length of the sentence, but is in fact simply  $O(1)$ , i.e. a constant. Covert-only movers thus have no impact on the asymptotic time complexity of MG parsing.

#### 4.2.6 Multiple Agree and DSMC

In section 3.3.5, we saw that the MGBank grammar includes the licensees -*pers*, -*num* and -*ep* in existential constructions involving a DP associate and an expletive *there* subject. These same features are also used in locative inversion and floating quantifier constructions in MGBank. In each case, a -*case* feature is split into multiple features which end up distributed across a subject and an associate, although the distribution of these feature across the two items is different for each construction. This approach is somewhat experimental and necessitated some special provisos with respect to DSMC in order to keep *k* as low as possible. The reason is that all three of these additional features must be allowed to trigger overt movement at various points in the MGBank

- (1) a. [<sub>*t*</sub><sub>*i*</sub> [kid-*bā*<sub>*i*</sub> ziya bištra] yoqsi] (Tsez)  
       [<sub>*t*</sub><sub>*i*</sub> [girl.ERG cow.ABS feed.INF] began]  
       ‘The girl began to feed the cow’.
- b. [<sub>*t*</sub><sub>*i*</sub> [kid-ber<sub>*i*</sub> babi-w-s xabar teq-a] y-oq-si] (Tsez)  
       [<sub>*t*</sub><sub>*i*</sub> [girl.II.DAT father.GEN story.III.INF hear.INF] begin.PAST.EVID]  
       ‘The girl began to hear the father’s story.’

Polinsky and Potsdam (2002) discuss a number of pieces of evidence in support of the claim that the overt DP in the above examples is in fact located in the embedded clause as indicated above, with the controller a null argument of the matrix clause. One such piece of evidence is that the case marking on the overt DP is that which is always taken by subjects of the embedded verbs in these constructions, regardless of whether or not they are embedded under other verbs. The verb for *to hear* in Tsez always takes a dative subject, for instance, while the verb for *to feed* takes an ergative one. Boeckx et al. (2010) also discuss a case of backward object control in Korean documented by Monahan (2003), in which case marking can similarly be used to show that the overt DP is the controllee.

From the perspective of Chomsky’s Copy Theory of Movement, Boeckx et al. (2010) argue that the MTC can account for backward control by assuming that in such constructions it is the lower copy of a moved item which is pronounced, rather than the higher copy. In the context of the formalism presented here, we can simply reinterpret this as the claim that backward control is an instance of covert control movement. While control is specified in the English MGBank grammar as being exclusively an overt movement type, therefore, it should be borne in mind that for other languages this would not be the case.



grammar, and must also be allowed to be simultaneously active with each other (although only two need ever be active at once).

In the case of *-epp*, the fact that this feature triggers overt movement will be clear from the discussion in section 3.3.5 where this feature was responsible for moving *there* to the surface subject position. We also saw that although *-pers* is licensed by a covert movement licenser, it must be checked overtly in existential constructions because it is placed immediately in front of the *-epp* feature on the below type for expletive *there*.

$$there :: d\{3\} -loc\{EDGE\} -pers -epp$$

Once a chain begins to move covertly, it cannot revert to moving overtly, hence it must keep moving overtly until the last overt licenser in a given sequence of movement steps is checked; as a result, *-pers* must be checked overtly in order to allow *-epp* to subsequently be checked by the *+EPP* licenser on the attracting head. The type that was used for the null [associator] head which links *there* to its DP associate and converts the latter's *-case* feature into *-num* is shown below.

$$[associator] :: d\{+INDEF\} = +case = d\{+3\} D -num$$

In this case, *-num* appears at the end of the licensee sequence, meaning that it can trigger covert movement unproblematically. However, consider the following examples.

(185) What dignity is there in that?

(186) How many issues are there?

Both of these examples feature a DP associate of expletive *there* which has undergone wh movement to the left periphery. Wh-movement is A'-movement and it therefore almost always follows any A-movements.<sup>21</sup> One consequence of this is that any associate wh DP will have to check *-num* *before* going on to check *-wh*. And indeed, MGBank uses the following special associator category for these constructions, in which *-wh* immediately follows *-num*, which must therefore be checked overtly.

$$[associator] :: d\{+INDEF\} = +case = d\{+3\} D -num -wh$$

<sup>21</sup>One exception to this is tough movement, which features A-to-A'-to-A movement in MGBank, a case of so-called *improper movement* - see section 6.4.2.1.

We can therefore see that all three of the features -pers, -num and -epp have the potential to impact the asymptotic time complexity of the parser, especially since they sometimes need to be active in the derivation at the same time as one another. The remainder of this section gives the details of a strategy which was adopted in order to avoid these features having any impact of the asymptotic time complexity of the parsers constructed for this project.

To begin, note that in each of the following three constructions that use -pers, -num and -epp, there are precisely two constituents involved in the subject-associate relation.

(187) **There** seems to be **a painting** on the wall. (existential)

(188) **On the wall** seems to be **a painting**. (locative inversion)

(189) **The paintings** seem to **all** be on the wall. (quantifier float)

187 features the expletive-associate relation that was discussed in section 3.3.5. In 188, the surface subject position of a raising verb appears to be occupied, somewhat surprisingly, by a prepositional phrase; the DP *a painting*, meanwhile, occurs in the same post-verbal position as it does in 187, and seems to have a similar (though not identical) associate relation to the PP as that between the associate DP and the expletive in 187. For instance, in either case, it is the DP associate rather than the syntactic subject which determines the number agreement on the verb.

(190) There seem(s)\* to be a painting of our ancestors on the wall.

(191) On the wall seem(s)\* to be a painting of our ancestors.

We saw in section 3.3.5 that there is evidence from locality that existential constructions involve some kind of A-movement. For example, this relation cannot cross a finite clause.

(192) \*There seems [ $CP_{fin}$  that a painting of our ancestors is on the wall].

The same also holds for locative inversion.

(193) \*[[On the wall seems]<sub>i</sub> [ $CP_{fin}$  that a painting of our ancestors is hanging  $t_i$ ]].

Furthermore, as noted in Bowers (2002), locative inversion cannot cooccur with expletive *there*, as example 194c below illustrates.

(194) a. There were several paintings on the wall.

- b. On the wall were several paintings.
- c. \*On the wall were there several paintings.<sup>22</sup>

That locative PPs should share some similarities with expletive *there* is arguably not surprising given that the latter plausibly derives diachronically from locative *there*. In MGbank, these similarities are captured by the fact that both constructions involve a finite T head entering into multiple checking relations with the subject and the associate. In the case of locative inversion, T checks -pers and -num on the DP associate (explaining the A-movement locality effects) and -epp on the PP which therefore occupies the surface subject position. The reason for the differential distribution of these features for locative inversion and existential constructions is that there are also some important empirical differences between them. For instance, the definiteness effect seen in existentials (195a) does not show up in locative inversion structures (195b).

- (195) a. There is a/\*the painting of our ancestors on the wall.  
 b. On wall is a/the painting of our ancestors.

In the MGbank grammar, this difference is explained by two factors: 1. the PP and its DP ‘associate’ do not form a constituent at any point in the derivation, unlike expletive *there* and its associate, hence there is no way to enforce the definiteness effect locally; and 2. only in existentials is the -pers feature stripped from the DP associate. In locative inversion, the PP is imbued with an -epp feature using a null [sbj] head, while the associate has its -case feature converted to a -pers -num sequence using a null [-sbj] head. As noted in section 3.3.5, MGbank adopts Radford’s (2004) proposal that -pers is the feature responsible for the definiteness property of DPs, and so the fact that this feature remains on the associate in locative inversion constructions in MGbank provides a competence-level explanation for why this associate can be definite.

From the similarities and complementary distribution of existentials and locative inversion, and given that both involve exactly one subject and exactly one associate, we can conclude that, as far as these two constructions are concerned, there will never be any need to allow more than two of the three features -pers, -num or -epp to be active in any derivation tree at any one time.

---

<sup>22</sup>The fact that 1 below is acceptable is irrelevant here, as this is clearly not locative inversion, but just a straight case of topicalization of the PP to the left periphery.

- (1) On the wall there were several paintings.

What about floating quantifiers which also use these features in the MGBank grammar? These clearly have very different properties to either expletive *there* or inverted PPs. However, we saw in section 3.2.7 that there is good evidence that floating quantifiers initially form a constituent with their DP antecedents, before the latter optionally breaks away and moves to the surface subject position. MGBank therefore adopts a doubling analysis of floating quantifier constructions which shares some similarities with that used for existentials (see Appendix A for the lexical categories involved).

Interestingly, floating quantifier constructions never mix with either existentials or locative inversion structures, in Standard English at least, as the following examples illustrate.

- (196) a. The paintings (all) seem to (all) be (all) on the wall.  
 b. There (all\*) seem to (all\*) be (\*all the) paintings (\*all) on the wall.  
 c. (\*All) on the wall (all\*) seem to (all\*) be (all the) paintings of our ancestors.

Quantifiers like *all* require a definite DP complement, and the associate of expletive *there* cannot be definite, which explains why *all* is barred from appearing in any position in 196b. The associate of a locative PP can be definite, explaining why in 196c *all* is permitted when it appears in an unstranded position with a definite determiner following it. However, when stranded, it must be c-commanded by its DP antecedent, which explains why all other positions for this *all* result ungrammaticality here.

The following examples may at first appear to undermine the claim that floating quantifiers cannot coexist alongside existentials and inverted locative PPs.

(197) There were several men [all waiting to see the doctor.]

(198) In the corridor were several men [all waiting to see the doctor.]

In 197 expletive *there* associates with *several men* which in turn appears to associate with the floating quantifier *all*. 198 is similar except that in place of *there* we have the PP *in the corridor*. However, the bracketed constituents above are plausibly analysed as reduced relative clauses (with null *wh* operators in spec-CP), so that 197 and 198 are the reduced versions of 199 and 200 below.

(199) There were several men [who were all waiting to see the doctor].

(200) In the corridor were several men [who were all waiting to see the doctor].

In these examples, the floating quantifier plausibly associates with *who*, rather than with *several men*. If the bracketed constituent in 197 and 198 is analysed as a reduced relative clause with a null counterpart of *who* in spec-CP, then *all* can again be allowed to associate with the wh operator, rather than with *several men*. This analysis is supported by the observation that *all* cannot in fact appear as a single constituent with a DP headed by *several*.

(201) All the/\*several men were waiting to see the doctor.

Given that these three constructions apparently need never mix with one another, and given also that each involves exactly one subject and exactly one DP associate, we can safely stipulate that *no more* than two of the three features -pers, -num and -epp may be active in the derivation at any one time. At the same time, we must also ensure that *up to* two of these features are permitted to co-occur (of course, two instances of precisely the *same* feature cannot be simultaneously active as this would violate classical SMC).

We can achieve this by adding our three features to both of the A-movement classes in the definition of DSMC from section 4.2. One feature can then be drawn from each class, meaning that -pers can co-occur with -num, for example, despite the fact that these features also appear together in a single class. Here then, is a second motivation for having two A-movement classes (the first was the need for the -self shield used in reflexive/reciprocal binding to not trigger intervention effects with -case). Because they appear in the class A-movement-1, these features will correctly trigger intervention effects for instances of A-movement triggered by -case or D.

As was noted in section 3.3.5, the -loc licensee on *there* and inverted locative PPs is used to ensure that these items are only licensed by intransitive verbs and progressive/copula *be* in Standard English (since only [intrans] little *v* and *be* have a +LOC! licenser). Because this feature also triggers overt movement, it too must be added to a DSMC feature class. It must also be permitted to co-occur alongside -num in existentials and alongside -pers for locative inversion. We will therefore treat it in the same way as these other features by adding it to both A-movement groups. We therefore now reformulate DSMC as follows.

**The derelativized Shortest Move Constraint (DSMC)** (final version): Two licensee features may both be active at the same time in the derivation only if they are drawn from different feature classes, where the feature classes are: A-movement-1 features (-case, D, -tough, -pers, -num, -epp, -loc), A-movement-2 features (-self, -num, -pers,

-epp, -loc), A'-movement-1 features (-top, -foc, -wh) and A'-movement-2 features ( $v\sim$ ,  $t\sim$ ,  $c\sim$ , -n).

DSMC thus continues to ensure that there can only ever be a maximum of 4 overtly moving chains in the derivation at any one time despite the introduction of the features -pers, -num, -epp and -loc which must be allowed to co-occur at certain points. As such, the introduction of these features has no impact on the worst case asymptotic time complexity of the formalism.

### 4.3 Containment-based locality constraints

There are, broadly speaking, two types of locality conditions which have been proposed in the TG literature since such conditions were first investigated in Ross (1967), Chomsky (1973) and Chomsky and Lasnik (1977). The first type are intervention-based locality constraints such as Rizzi's Relativized Minimality, the Shortest Move Constraint, Minimal Link Condition and Attract Closest Principle of Chomsky (1995). As we have seen, in the present framework these types of intervention-based locality constraints are approximated by SMC and DSMC. As was noted in section 3.2.10, intervention-based locality constraints involve the configuration shown below, in which B blocks the formation of a (direct) dependency between A and C.

(202)                            [... A ... [... B ... [... C ... ] ] ]

As we have seen, in order for B to block the dependency between A and C, B is required to be of the same or similar type to either A or C and must generally c-command C.

The second class of locality conditions are what Gärtner and Michaelis (2007) refer to as *containment-based* locality constraints. These involve the following configuration.

(203)                            [... A ... [<sub>B</sub> ... C ... ] ]

Here, the formation of a dependency between A and C is blocked if C is contained within a phrase of type B. In this case, B usually need not be of a similar type to ei-

ther A or C,<sup>23</sup> but can in principle be any type of syntactic phrase, defined in either functional or categorial terms. In the following sections we will consider a number of containment-based constraints on movement which have been proposed in the literature, and will see how these have (partially) been incorporated into MGbank in order to reduce the parser's search space and improve the descriptive adequacy of the grammar.

### 4.3.1 Condition on Extraction Domains

Huang (1982) (building on observations made in Ross (1967)) identified an important constraint on movement which is stated informally in Radford (2004, page 218) as follows.

(204) **Condition on Extraction Domains (CED)**

Only complements allow material to be extracted out of them, not specifiers or adjuncts.

CED actually embodies what are arguably best treated as two separate constraints: the Specifier Island Constraint (SpIC) and the Adjunct Island Constraint (AIC). These constraints are intended to describe the following contrasts.

- (205) a. He took [pictures of *who*]?  
b. *Who*<sub>i</sub> did he take [pictures of *t<sub>i</sub>*]?
- (206) a. He was angry [that you ate *what*]?  
b. *What*<sub>i</sub> was he angry [that you ate *t<sub>i</sub>*]?
- (207) a. [Pictures of *who*] caused a scandal?  
b. \**Who*<sub>i</sub> did [pictures of *t<sub>i</sub>*] cause a scandal?
- (208) a. He was angry [when you ate *what*]?  
b. \**What*<sub>i</sub> was he angry [when you ate *t<sub>i</sub>*]?

In each of the above sentence pairs, the (a) example features an echo question in which a wh word remains in situ and the sentence is perfectly acceptable (with the indicated intonation). The bracketed constituents in both 205 and 206 are complements and, as shown by 205(b) and 206(b), extraction of the wh word to the left periphery also yields a grammatical sentence in these cases. The situation is very different with

---

<sup>23</sup>One exception is Chomsky's (Chomsky (1964)) A-over-A condition, in which B must be of the same type as C.

207 and 208, however, where extraction from a specifier (207(b)) or adjunct (208(b)) results in a degraded sentence. Thus extraction from specifiers and (clausal) adjuncts is generally disallowed in English, in contrast to extraction from complements.

One way to improve the efficiency of parsing is to narrow the search space of the parser by allowing it to enforce locality constraints such as these. At the same time, it is important to make sure that the grammar does not become so constrained that it begins to undergenerate too severely. There is always a delicate balance to be struck. In the following two sections, we will examine the role that SpIC and AIC each play in the MGBank grammar.

#### 4.3.1.1 The Specifier Island Constraint

The Merge rules from section 3.2.6 of this chapter already enforce a version of SpIC which Kobele and Michaelis (2011) refer to as  $\text{SpIC}_{mrg}$ .  $\text{SpIC}_{mrg}$  only blocks extraction from specifiers created by Merge, not those created by Move. None of the Merge rules combining a specifier with its head from section 3.2.6 allowed for the case where the selected expression contains any movers, thus ensuring that  $\text{SpIC}_{mrg}$  is never violated. However, we shall see below that it ultimately proved infeasible to enforce this condition for all constructions in MGBank. In particular, those involving an overt anaphor or associate, such as reflexive binding, existentials and floating quantifier constructions, required some mechanism for circumventing the effects of  $\text{SpIC}_{mrg}$ .

The formal consequences of enforcing SpIC have been investigated quite extensively within the MG framework (Michaelis, 2004; Gärtner and Michaelis, 2005; Michaelis and Kobele, 2005; Gärtner and Michaelis, 2007; Kobele and Michaelis, 2009; Kanazawa et al., 2011). For example, Kanazawa et al. (2011) showed that MGs with both SMC and SpIC can generate only well-nested MCFLs, which exclude the MIX language, making them strictly mildly context sensitive in Joshi's sense, though still more expressive than the TAG-class grammars.<sup>24</sup> Kobele and Michaelis (2011), meanwhile, show that the same result is true even for the weaker  $\text{SpIC}_{mrg}$ . This is an interesting formal result because it aligns with the claim in Chomsky (2008) that extraction from passive subjects is far more acceptable than extraction from active ones.

- (209) a. \*[Of which car]<sub>i</sub> did [the driver t<sub>i</sub>] cause a scandal?  
 b. [Of which car]<sub>i</sub> was [the driver t<sub>i</sub>]<sub>j</sub> awarded t<sub>j</sub> a prize?

<sup>24</sup>On the other hand, Kobele and Michaelis (2009) show that MGs without SMC, but with SpIC, are Turing complete.



(Chomsky, 2008, page 14)

If, as TG analyses standardly assume, passive subjects originate as objects, then the *wh* constituent in 209b can be extracted while the larger DP containing it is still in its base object position, thereby circumventing  $\text{SpIC}_{mrg}$ . As noted by Stabler (2013b) (where  $\text{SpIC}_{mrg}$  was shown to be advantageous for top-down MG parsing) several other linguists have also argued that the so-called freezing effect obtained by banning movement out of specifiers is probably too restrictive (see, e.g., Sauerland (1999), Koopman and Szabolcsi (2000), Collins (2005), Abels (2007), all cited in Stabler (2013b)).

In an earlier publication (Torr and Stabler, 2016), it was stated that the MGbank grammar conformed to SpIC (which was actually  $\text{SpIC}_{mrg}$ ) and this was indeed the case at that time. Despite the attractiveness of keeping the present formalism firmly within the class of mildly context sensitive formalisms, however, strictly enforcing  $\text{SpIC}_{mrg}$  did not ultimately prove to be possible for describing certain long-distance dependencies where both the antecedent and the anaphor (or associate) are overtly expressed.

We have already seen a number of these constructions earlier in this chapter. One example was the binding of reflexives discussed in section 4.2.4. Recall that in that analysis, the reflexive and its antecedent begin their derivational lives by being combined into a single constituent. This was achieved using a null [self] morpheme which first takes the antecedent as its complement. The resulting constituent is then selected as a complement by a reflexive anaphor such as *himself*, which has its own set of A-movement licensees. This doubled constituent is then merged as the argument of some verb, after which both the [self]+antecedent complex and the reflexive begin to move to check their various features. The problem with this for  $\text{SpIC}_{mrg}$  is that the doubled constituent may well be First Merged into a specifier position. This is true wherever an ECM verb selects a transitive/unergative clausal complement, for instance. For instance, in 210 below the doubled constituent originates in the spec-vP position of the *help* clause.

(210)  $Jack_i$  believed [ $himself\ t_i$ ] $_j$  to have [ $_{vP}\ t_j$  helped Mary].

Because the doubled constituent is a specifier (in both in its base and surface positions), moving *Jack* out of this constituent violates  $\text{SpIC}_{mrg}$ . Note furthermore that even if we were to adopt Hornstein's (2001) suggestion that the reflexive adjoins to the antecedent, rather than taking it as its complement, we would still have the problem

that the reflexive would then need to move out of the doubled constituent containing the antecedent, which would again violate  $\text{SpIC}_{mrg}$ .

Another construction which we have seen that violates  $\text{SpIC}_{mrg}$  is the floating quantifier construction. Although the precise details of the MGBank analysis of this construction were not yet described, we saw in section 3.2.7 that there is good evidence that the stranded quantifier *all* in 211 below begins its derivational life by forming a constituent with the antecedent subject, before the subject undergoes movement away from the quantifier to the surface subject position.

(211) The workers have all eaten their lunch.

As with the binding of reflexives, MGBank treats this construction as involving a doubled constituent consisting of the subject and the quantifier. The quantifier then exits this doubled constituent, moving either to the specifier of a second outer vP shell layer, or to the specifier of a higher auxiliary (accounting for sentences such as *the workers must all have eaten their lunch*), before the remnant doubled constituent containing the subject moves to the surface subject position. In many cases, the doubled constituent will be First Merged into spec-vP, as indicated below, meaning that we once again have a violation of  $\text{SpIC}_{mrg}$ .

(212) [<sub>t<sub>i</sub></sub> The workers]<sub>j</sub> have [<sub>vP</sub> all<sub>i</sub> [<sub>vP</sub> t<sub>j</sub> eaten their lunch]].

The fact that  $\text{SpIC}_{mrg}$  does not hold for these examples could simply be down to the particular analyses themselves, of course. However, in the MGBank formalism the only possible way to form long-distance dependencies in the syntax between two constituents which both have overt phonetic content is to adopt this doubling approach. I do not think that there is therefore any way at present to avoid these problems, assuming we are determined to capture these dependencies in the syntax.<sup>25</sup> It may be possible to extend the formalism without increasing its expressive power so that these constructions could be adequately described without the need to violate  $\text{SpIC}_{mrg}$ . This is left for future research.

For standard cases of movement out of specifiers, however,  $\text{SpIC}_{mrg}$  does seem to be fairly robustly applicable, at least for English. Even though we cannot reduce

<sup>25</sup>As was noted in section 3.2.7, one of the advantages of the MG formalism is the fact that the syntax already encodes a considerable amount of the predicate argument structure, which should make it relatively straightforward to write a compositional semantics for MGs. Capturing as many long-distance dependencies such as control and reflexive/reciprocal binding directly in the syntax is therefore advantageous in this respect. A further advantage is that including such dependencies in the syntax allows the statistical parsing model to condition over them.

the weak expressive power of the formalism by enforcing it as a strict constraint in MGBank, then, we can still profitably reduce the parser's search space by enforcing it for the many cases where it does hold.

In MGBank this is achieved using a selectional property *EDGE* which is added to specific licensees in the lexicon. Any moving chain whose active licensee bears an *EDGE* property will be allowed to escape a specifier island. In order for a derivation to be allowed to continue when a specifier is Merged into the main structure, therefore, one or more of the following three conditions must be met: 1. the specifier has no moving chains; 2. ATB movement can apply as described in section 3.3.7.2; or 3. all of the specifier's moving chains must have an *EDGE* property associated with their active licensee.

In section 4.2.4 it was noted that ECM examples such as 213 below were problematic for the doubling analysis of reflexive binding because *Jack* is required to escape from the doubled constituent which is base generated in spec-vP.

(213)  $Jack_i$  expects  $[himself\ t_i]_j$  to  $[_{vP}\ t_j\ [_{v'}\ win]]$ .

We can now solve this issue by adding an *EDGE* property to the -self licensee of the [self] head involved in this construction, as shown below (at this point, the reader should refer back to fig 4.9 and satisfy themselves that this will correctly circumvent the effects of  $SpIC_{mrg}$  for reflexive objects of ECM verbs).

$$[self] :: d\{x\} = +case\{y\}\ self\{x\}\ -self\{EDGE\}\ D\ -case\{y\}$$

Similarly, floating quantifiers must be allowed to escape from the doubled quantifier+antecedent constituent in order to escape the spec-vP island in examples such as 214 below.

(214)  $[t_i\ the\ boys]_j$  must  $all_i$  have been  $[_{vP}\ [_{v'}\ t_j\ helping]]$ .

For this reason, floating quantifiers in MGBank include an *EDGE* property on their -epp feature. The MG type for the floating quantifier *all* in MGBank is given below.

$$all :: q\{FLOAT.NOMOD.PL.SG\}\ -epp\{EDGE.FLOAT\}\ -num\{OVERT.PL.SG\}$$

We will now formalise this escape mechanism by adding the rules in fig 4.14 to the grammar, where each  $\alpha_i^e$  is some moving chain whose active licensee is decorated with the property *EDGE*.

$$\frac{[t_l, t_h, t_r; x, \alpha_1^e, \dots, \alpha_k^e] \quad [s_l, s_h, s_r; =x \gamma]}{[t_l t_h t_r s_l, s_h, s_r; \gamma, \alpha_1^e, \dots, \alpha_k^e]} (\text{merge\_esc\_1})$$

$$\frac{[t_l, t_h, t_r; x \delta, \alpha_1^e, \dots, \alpha_k^e] \quad [s_l, s_h, s_r; =x \gamma]}{[s_l, s_h, s_r; \gamma, t_l t_h t_r; \delta, \alpha_1^e, \dots, \alpha_k^e]} (\text{merge\_esc\_2})$$

Figure 4.14: Sub-functions of Merge allowing for escape from base generated specifier islands.

Notice that the type identifier for the selecting expression here is  $:$  in these rules. This means that this escape mechanism is not available for conjunct specifiers (the type identifier for a coordinator expression selecting a leftward conjunct is  $\bar{}$  - see section 3.3.7.1).

In the remainder of this thesis we will simply refer to SpIC, which unless otherwise stated is a shorthand for the version of SpIC<sub>mrg</sub> with the escape mechanism described here. Note that SpIC<sub>mv</sub>, the version of SpIC which blocks extraction from specifiers created by movement, is not enforced here at all.

#### 4.3.1.2 The Adjunct Island Constraint

The formal consequences of enforcing the Adjunct Island Constraint were investigated in Gärtner and Michaelis (2007) who conjectured that, as with SpIC, adding AIC to an MG without SMC does not constrain weak expressivity. Gärtner and Michaelis (2003) show that MGs can even be extended with a *late adjunction* operation without affecting either strong or weak generative capacity (although doing so does increase what they refer to as the *derivational generative power* of the formalism). Late adjunction is an operation which was proposed within mainstream TG by Lebeaux (1988) as a way of accounting for the observation, originally made in Van Riemsdijk and Williams (1981), that under certain circumstances R-expressions contained within an adjunct appear able to escape Principle C violations. Gärtner and Michaelis (2007) subsequently showed that adding such an operation to an MG which also includes movement to adjoined positions via scrambling or extraposition appears to allow the grammar to circumvent the constraining effects of SMC, but that strict enforcement of AIC guarantees that this cannot happen and that weak expressive power is not thereby increased.

The MGbank grammar does not include a treatment of anti-locality constraints such as Principle C (or Principle B) of Chomsky's (1981) Binding Theory as these are

fundamentally different in nature from standard locality effects such as Principle A which was enforced in section 4.2.4 by (D)SMC. For this reason, late adjunction was not included as an operation in MGBank. As with SpIC, there were cases where strict enforcement of AIC proved too restrictive. Consider the following examples.

(215) a. There's some [price<sub>i</sub> [at which t<sub>i</sub>]<sub>j</sub> we'd stop bidding t<sub>j</sub>].

(216) a. Which price would you stop bidding [at t<sub>i</sub>]?

(217) a. He also had to fight [harder than his partner did] for credibility.

b. He also had to fight [harder t<sub>i</sub>] for credibility [than his partner did]<sub>i</sub>.

Example 215a is taken from the Penn Treebank. Under MGBank's promotion analysis of restrictive relative clauses (discussed in section 4.2.1), the head noun must move to the left periphery of the relative clause independently of the *wh* phrase. Given that the phrase *at which price* is most plausibly analysed as an adjunct, the head noun must be allowed to move out of this adjunct in violation of AIC. Of course, this analysis of relative clauses will not be uncontroversial, and there are certainly other analyses which would not require this violation (these would include the promotion analysis of Kayne (1994), but also non-promotion analyses in which the head noun is generated outside the relative clause). The analysis of the related example 216a, however, would be far less controversial. Here the *wh* phrase and the noun clearly vacate the adjunct PP as a unit on their way to the left periphery, thereby stranding the preposition and violating AIC.

Example 217b is also taken from the Penn treebank, and there are many other cases like this, where the complement of a comparative adverb or adjective undergoes extraposition. The phrase *harder than his partner did* in 217a is arguably most plausibly analysed as an adjunct, and so extraposition from within this adjunct in 217b is again in violation of AIC.

A less restrictive version of the AIC that is sometimes adopted is the following.

(218) **The Adjunct Island Constraint**

Nothing may be moved out of a *clausal* adjunct.

Here, AIC is restricted to only applying to clausal adjuncts. This allows for the violations of the stronger AIC exhibited above, while also correctly blocking 219, 220 and 221 below (taken from Szabolcsi (2006)).

(219) \*<sub>[PP About which topic]<sub>i</sub></sub> did you leave [because Mary talked t<sub>i</sub>]?

(220) \*<sub>[DP Which topic]<sub>i</sub></sub> did you leave [because Mary talked about t<sub>i</sub>]?

(221) \*<sub>[AdvP How]<sub>i</sub></sub> did you leave [because Mary behaved t<sub>i</sub>]?

This weaker, clausal version of the AIC is implemented in MGbank in the following manner. When an adjunction operation is attempted, assuming that the adjunct contains movers and that ATB movement cannot apply to circumvent the adjunct island (see section 3.3.7.2), the system will check to see whether the head chain of the adjunct expression (which originates from the overt or null adjunctizer head) has previously had any =c/c= or =t/t= selector features checked. If it has, then the adjunctizer selected a clausal dependent meaning that the adjunct is a clause and so the operation is blocked. This procedure exploits the dotted feature mechanism introduced in section 3.3.7.4 (and borrowed from Kobele (2008)), which allows the system to see the features which have already been checked and ‘deleted’ on a given expression.

It must be stated that the situation with adjunct islands is considerably more nuanced and complex than the simple distinction between clausal and non-clausal adjuncts made above allows for. For example, observe that extraction of a PP or AdvP from a gerund clause is in many cases also quite bad.

(222) a. You left [without talking about *which* topic]?!

b. \*<sub>[About which topic]<sub>i</sub></sub> did you leave [without talking t<sub>i</sub>]?

(223) a. You left [without behaving *how*]?!

b. \*<sub>[How]<sub>i</sub></sub> did you leave [without behaving t<sub>i</sub>]?

In order to block examples such as 222b and 223b, the MGbank grammar treats adverbial gerund adjunct clauses as projecting CPs, rather than DPs as was the case for the gerund complements in transitive case positions discussed in section 4.2.3. A complication with this, however, is that extraction of a DP from an adverbial gerund clause leads to less degradation in acceptability.

(224) ?<sub>[Which topic]<sub>i</sub></sub> did you leave [without talking about t<sub>i</sub>]?

It seems that to some extent, the tense properties of the clause play a role in determining whether or not AIC applies. This is shown by the fact that extraction of a DP from an infinitival purposive adjunct clause is sometimes perfectly permissible.

(225) <sub>[Which book]<sub>i</sub></sub> did you go to the library [to borrow t<sub>i</sub>]?

Even the clausal version of the AIC is clearly too restrictive, therefore. However, of the 804 trees of the PTB which I hand annotated, none required violation of clausal AIC, though several required violations of the strict version of AIC. For current purposes, therefore, the clausal distinction appears to strike the right balance between the competing demands of the avoidance of overgeneration and undergeneration.

### 4.3.2 The Coordinate Structure Constraint

Ross (1967) proposed the Coordinate Structure Constraint (CSC), which we can state informally as follows.

(226) **The Coordinate Structure Constraint (CSC)**

1. Conjuncts cannot be extracted.
2. Constituents cannot be extracted from within conjuncts, except in across-the-board fashion.

Part 1 of the CSC is illustrated by 227b and 227c, while part 2 is exemplified by 228b and 228c; example 228d shows that across-the-board extraction from within conjuncts is possible.

- (227) a. I bought some books and some records.  
 b. \*[Which books]<sub>i</sub> did you buy t<sub>i</sub> and some records?  
 c. \*[Which records]<sub>i</sub> did you buy some books and t<sub>i</sub>?
- (228) a. I bought some [pictures of Madonna and some recordings of Bowie].  
 b. \*Who did you buy [some pictures of t<sub>i</sub> and some recordings of Bowie]?  
 c. \*Who did you buy [some pictures of Madonna and some recordings of t<sub>i</sub>]?  
 d. Who did you buy [some pictures of t<sub>i</sub> and some recordings of t<sub>i</sub>]?

The CSC is in general a very robust constraint both in English and cross-linguistically, and was in fact described by Postal (1998) as “the most problem-free syntactic constraint ever discovered” (although arguable exceptions do exist in English to part 2 - see below). The MGBank formalism therefore enforces the CSC rigidly. This is done by allowing the parser to refer to the  $\bar{\bar{\alpha}}$  or  $\bar{\alpha}$  type identifier appearing on conjuncts (see section 3.3.7). Whenever a complement or specifier is merged with a governor which is a coordinator complex, the system will abort the parse if that complement or specifier has a licensee feature behind its selectee (or if it has a persistent D feature), thus enforcing part 1.

Part 2 is enforced at the point at which the specifier conjuncts are merged into the structure, in order to allow for ATB movement. As described in section 3.3.7.2, if the movers inside the incoming specifier conjunct exactly match those inside the main structure, the matching movers in the specifier will be deleted (equivalent to unifying the matching movers) and the derivation will continue, otherwise it will be aborted. This enforces part 2 of the CSC.

There are some well-known exceptions to part 2 of the CSC which occur in so-called asymmetrical coordination structures, where there is some causal, temporal or other relation between the two conjuncts. In such cases extraction can often occur from either conjunct, as exemplified by 229 and 230 below (examples taken from Zhang (2010)).

(229) [How much wine]<sub>i</sub> can you drink t<sub>i</sub> and/but still stay sober?

(230) [Which knee]<sub>i</sub> did Terry [run in these shoes and hurt t<sub>i</sub>]?

Zhang (2010, page 139) argues that such cases support her thesis that the CSC should be abandoned. We do not pursue this strategy here, however, as the CSC is in general very useful for constraining the parser's search space, and in the vast majority of cases it has no adverse effects. There is, in any case, some evidence that the versions of *and* and *but* found in asymmetric coordination deserve a separate treatment from that of standard coordinators (see Postal (1998, pages 56-60) for an overview). For example, asymmetric coordinators cannot be replaced by disjunctive *or* (231), asymmetric coordination must involve minimal VPs, rather than clauses or VPs containing an auxiliary (232), the correlative focus coordinator *both* cannot co-occur with this type of *and* (233), and of course, there is an obligatory causal, temporal or other semantic relation holding between the two conjuncts which is not the case for standard coordination.

(231) The cheese<sub>i</sub> which Frank went to the store, bought t<sub>i</sub>, went home, and/\*or gave t<sub>i</sub> to Greta. (example from Postal (1998, page 57))

(232) \*[How much wine]<sub>i</sub> can you drink t<sub>i</sub> and can still stay sober?

(233) \*[How much wine]<sub>i</sub> can you both drink t<sub>i</sub> and still stay sober?

The MGBank formalism does provide a mechanism for constituents to escape specifier islands (see section 4.3.1.1), and non-coordinator categories permit extraction from their complements in any case. Hence the examples of asymmetric coordination



given here are in principle generatable using separate coordinator heads not marked as coordinators,<sup>26</sup> although one would still need to take into account the fact that these items do share some formally significant properties with standard coordinators, perhaps most pressingly the fact that they allow for recursion with respect to the left conjuncts (231), meaning that their =x selector would still need to be allowed to optionally persist assuming the left conjuncts were treated as specifiers, and not adjuncts for instance.<sup>27</sup>

### 4.3.3 The Right Roof Constraint

Like other types of movement, rightward movement appears to be subject to certain locality restrictions. In particular, it has been proposed that it is clause bounded, a condition known as the Right Roof Constraint (Ross, 1967). This clause boundedness is illustrated by the following contrast (irrelevant traces of A-movement are omitted here).

- (234) a. [<sub>CP</sub> [<sub>CP</sub> That [a complaint about his behaviour] was made yesterday] is unfortunate].
- b. [<sub>CP</sub> [<sub>CP</sub> That [a complaint t<sub>i</sub>] was made yesterday [<sub>PP</sub> about his behaviour]<sub>i</sub>] is unfortunate].
- c. \* [<sub>CP</sub> [<sub>CP</sub> That [a complaint t<sub>i</sub>] was made yesterday] is unfortunate [<sub>PP</sub> about his behaviour]<sub>i</sub>].

In example 234b, the PP does not move outside of the embedded CP, and the sentence is grammatical. This contrasts with the situation in 234c, where the PP moves across the embedded CP into the matrix clause, resulting in ungrammaticality. As a further example of the effects of the RRC, consider the following.

- (235) a. He will claim that I read a passage from Ulysses yesterday.
- b. He will claim that I read a passage t<sub>i</sub> yesterday [from Ulysses]<sub>i</sub>

<sup>26</sup>Given that these items are strongly conditioned by semantics, we would look to the probability model to constrain their distribution so that the parser does not suddenly start allowing extraction from all coordinate complexes; for instance, the model could learn that when *and* is followed by *then* or *still* there is a higher probability that it is an asymmetric coordinator than otherwise.

<sup>27</sup>MGBank only enforces the Adjunct Island Constraint for clauses (see 4.3.1.2), not verb phrases, so extraction from adjunct VP conjuncts would be allowed. However, the fact that if only one coordinator appears in asymmetric coordinate complexes, it must immediately precede the final conjunct, suggests that this is indeed a basic Xbar configuration with the rightmost conjunct a complement and all leftward conjuncts specifiers.

- c. He claimed that I will read a passage from Ulysses yesterday.
- d. \*He claimed that I will read a passage  $t_i$  yesterday [from Ulysses]<sub>*i*</sub>

In 235a, the adverb *yesterday* is necessarily construed as modifying the embedded clause. This means that the PP *from Ulysses* can be extraposed without needing to cross a clause boundary, and the resulting sentence 235b is fine. 235d, on the other hand, is clearly degraded, and this arguably because the adverb in 235c can only modify the matrix clause, which forces rightward movement of the PP to cross a clause boundary in this instance.

It should be noted that whether or not the RRC is a genuine syntactic constraint has been contested (see, e.g. Gazdar (1981)). However, whether or not it is ultimately part of syntax proper or whether it is a processing constraint<sup>28</sup> (which has perhaps become grammaticalised in many cases), its effects are real and can thus profitably be incorporated into the grammar in order to constrain the search space with very little loss in empirical coverage.

We can approximate the RRC for all CP clauses by making the following stipulations with respect to the MGBank grammar.

1. Rightward movement licensees never persist, i.e. they are always deleted when checked.
2. If rightward movement can apply within a given XP, it must apply before that XP is Merged with any other constituent (rightward movement and adjunction may freely intersperse, however), unless the Merge operation involves a coordinator as the selecting head.
3. For all rightward movement licensee features  $x\sim$ ,  $x$  must be drawn from the following set of major clausal categories which appear in (virtually) all clauses:  $\{c, t, v\}$ .

Restriction 1 ensures that rightward movement cannot escape the RRC by moving successive cyclically through intermediate checking positions which fail to delete the  $x\sim$  licensee. Restriction 2 avoids the situation where Merge bleeds rightward movement by checking and deleting the  $x$  selectee before the rightward movement has taken place. Finally, Restriction 3 is necessary as long as we assume (contra Cinque (1999))

<sup>28</sup>See, e.g., Grosu (1973), Berwick and Weinberg (1986), Fodor (1978, 1979) and Frazier (1985), Rochemont (1992).

that it is not the case that all clauses contain all possible clausal projections. For example, MGbank uses a negP projection just for clauses featuring negation. If we were to allow a neg $\sim$  licensee, therefore, then any constituent containing this as its first feature could simply keep moving through an unbounded number of non-negated clauses.

The only clausal rightward movement licensees used in MGbank are therefore the following:  $v\sim$ ,  $t\sim$ ,  $c\sim$ , and since VP, TP and CP are present in most clauses, this ensures the clause boundedness of rightward movement for these clauses. There are exceptions, however. For example, as in MP, in the MGbank grammar the complement clauses of ECM and raising verbs are bare TPs lacking the CP layer;  $c\sim$  licensees can at present be used to escape the RRC in MGbank. Whether or not this is the correct result is unclear to me; 236b, in which the PP *from my diary* escapes the TP clause, seems marginally ok to my ears.

- (236) a. He believed me [ $_{TP}$  to be reading a passage from my diary] with every inch of his being.  
 b. ? He believed me [ $_{TP}$  to be reading a passage  $t_i$ ] with every inch of his being [from my diary] $_i$ .

Superficially similar object control verbs do take full CP complements in MP and MGbank, on the other hand, and 237b below, in which the PP *this time* escapes a CP clause, is to my ears only slightly more degraded than 236b.

- (237) a. He persuaded me [ $_{CP}$  to read a passage from my diary] with his charm.  
 b. ?? He persuaded me [ $_{CP}$  to read a passage  $t_i$ ] with his charm [from my diary] $_i$ .

In Baltin's (1981) classic analysis of rightward movement, VP and TP are the two potential landing sites for rightward movement (this is forced by his theory of generalised subjacency, which is stricter than the RRC and is not adopted here). Almost all of the [extraposer] categories in MGbank use either  $v\sim$  or  $t\sim$  licensees. There are, however, certain constructions which arguably require extraposition to CP, one of which is illustrated below.

- (238) a. [ $_{CP}$  [ $_{CP}$  Who is  $t_i$ ] [ $_C$  and] [ $_{CP}$  who should be  $t_i$ ] [ $_{VP}$  making the criminal law here] $_i$ ]?

In MGbank, right node raising constructions such as this are analysed as across-the-board rightward movement (here, of a  $vP$ ). The two conjuncts are clearly CPs

because they each contain a preposed *wh*-word and such items are standardly analysed as occupying *spec-CP*. The only possible landing site for the extraposed phrase that is external to both of the two conjuncts is thus the outer *CP*,<sup>29</sup> and *MGbank* therefore includes  $c\sim$  in its inventory of licensees to cover such cases. Notice, however, that in order to arrive at this outer *CP*, the two instances of the extraposed phrase also have to bypass the inner *CP* conjunct phrases which dominate them. This means that in the case of coordination, Merge must be allowed to bleed rightward movement, so that the *c* selectees of the conjuncts can be eliminated before they attract the rightward moving *vPs*. This is the reason why stipulation 2 above contains an exception clause for coordinator heads.

Finally, note that although stipulation 2 prevents Merge from applying before rightward movement in the non-coordination case, it does not, at present, prevent adjunction from freely interspersing with rightward movement in *MGbank*. Adjunction does not eliminate the *x* licenser for rightward movement and so allowing it to freely interperse with rightward movement in this way does not affect our implementation of the *RRC*. The main reason for allowing the two operations to interleave was that rightward movement appears to be able to proceed to positions either outside or inside of temporal adjuncts which canonically appear adjoined to *TP* in *MGbank*. For instance, both 239a and 239b are perfectly acceptable.

- (239) a. She introduced  $t_i$  to the guests [the famous detective from Belgium]<sub>*i*</sub> *yesterday*
- b. She introduced  $t_i$  to the guests *yesterday* [the famous detective from Belgium]<sub>*i*</sub>

There are several other possible explanations of this interleaving effect, of course, such as that the temporal adjunct may adjoin to some other projection below *TP* (such as *vP*) in 239b, or that rightward movement actually targets *CP* (as proposed for instance in McCloskey (1999)) rather than *TP* here. As things stand, however, some of the *MGbank* trees rely on the assumption that rightward movement and adjunction may freely interleave within a given *XP*. Note that this represents the only exception to the Move-over-Merge constraint proposed in section 4.2.4.

<sup>29</sup>As we saw in section 3.3.7.1, the rightmost conjunct of a coordinate structure is analysed here as a complement of the coordinator head while any left conjuncts are specifiers; the coordinator head itself shares the same selectee category feature as its conjuncts.

#### 4.3.4 Phases, Successive Cyclic A'-movement and ?-type suicidal licensors

Chomsky (2000, 2001, 2008) argues that rather than the computational system waiting until the end of the entire derivation to send the tree to the CI and AP interfaces, derivations proceed cyclically, with much smaller chunks of structure undergoing interpretation; at this point, these chunks are closed off for further computation. More specifically, Chomsky proposes that CP and transitive vP, which he dubs v\*P, are *phases*, and that the once the head of the phase (C or v\*) is Merged into the structure and its features have been valued and deleted, the complement of the phase head (TP or VP) is transferred to the interfaces and becomes computationally inert. The complement of the phase head thus becomes an opaque domain as far as Move or Agree operations are concerned.

This may at first appear to create a problem for examples such as 240 below, in which a wh-mover has been extracted out of a transitive clause into a higher clause, crossing three phase barriers along the way.

(240) [[Which book]<sub>i</sub> did you [<sub>v\*P</sub> say [<sub>CP</sub> that Pete [<sub>v\*P</sub> borrowed t<sub>i</sub> ?]]]]

Chomsky argues, however, that the edge of the phase, which consists of the phase head and any specifiers it has, does remain active for further computation. Chomsky therefore proposes the *Phase-Impenetrability Condition*.

(241) *Phase-Impenetrability Condition*

In a phase  $\alpha$  with head H, the domain of H is not accessible to operations outside  $\alpha$ , only H and its edge are accessible to such operations.

The phase edge thus provides an escape hatch for a moving element, such as a wh constituent. 240 above is therefore more accurately represented as 242 below, with intermediate traces marking the successive cyclic movement of the wh constituent through each phase edge.

(242) [[Which book]<sub>i</sub> did you [<sub>v\*P</sub> t<sub>i</sub> say [<sub>CP</sub> t<sub>i</sub> that Pete [<sub>v\*P</sub> t<sub>i</sub> borrowed t<sub>i</sub> ?]]]]

If, on the other hand, one of the phase edges is already occupied by an A'-moved constituent, the escape hatch will not be available. This accounts for the ungrammaticality of 243, which features movement across a CP phase whose specifier is already occupied by *why*.

(243) \*Which book did you wonder [<sub>CP</sub> why Pete borrowed]?

Chomsky (2001a, page 13) argues that phases ‘minimise search’ between a probe and a goal and therefore result in the computational system having a ‘bounded memory load’ (Chomsky, 2001b, page 15). Furthermore, while early Minimalism eliminated D-Structure and S-Structure representations, Chomsky (2001:15, 2004:151) argues that phases allow us to go a step further and also dispense with LF as a separate cycle. The last vestige of S-Structure is also thereby eliminated, because it is no longer possible to perform movement covertly after Spell-Out has transferred all phonological features to PF, because Spell-Out is now occurring cyclically throughout the course of the derivation rather than occurring once between separate overt and covert cycles. This results in a single, overt cycle, with the long-distance operation Agree stepping in to cover much of the empirical ground previously treated in terms of covert/LF movement.

Phases Theory has been highly influential within MP, receiving widespread acceptance in the field. At the same time, that acceptance has not been unanimous and phases are considered controversial by a substantial minority of Minimalists. For example, Chomsky attempts to justify his choice of v\*P and CP as phases by suggesting that these are isolable units, ‘relatively independent in terms of interface properties..the closest syntactic counterpart to a proposition: either a full verb phrase in which all *O*-roles are assigned, or a full clause including tense and force,’ (Chomsky, 2000, page 106), but this approach has been criticised from a number of perspectives. Epstein and Seely (2006, pages 61-62) point out, for instance, that under this definition, intransitive vP should also be a phase because here too all *O* roles are assigned, and indeed Legate (2002) has shown that intransitive vP passes many of the proposed diagnostics for phasehood. Epstein et al. (2015, page 79) also observe that even in transitive clauses TP also has all *O* roles assigned, so that Chomsky’s proposal implies that “‘full argument structure’ must be a *translexical notion*.” Grohmann (2000), Bošković (2002), Abels (2003) and others have also pointed out that given Chomsky’s claim that what is transferred to the interfaces is not the phase itself but the complement of the phase head, it should arguably be the complement of the phase head (i.e. TP or VP) which is ‘relatively independent in terms of interface properties,’ rather than the phase itself. In sum, Chomsky’s choice of v\*P and CP and the phases in the clausal domain has been argued to boil down to a stipulation.

Chomsky’s (2000) original empirical argument for phases involved associating them with lexical subarrays in order to account for certain exceptions to his Merge-over-Move principle. However, we shall see in section 4.7.3 that these arguments do

not apply here given MGBank's adoption of Epstein and Seely's (2006) proposal that ECM/raising *to* checks no A-features (which as we shall see we are forced to do as a consequence of our decision to treat control as an instance of A-movement following Hornstein (2001)). Boeckx (2009, page 59) notes that in any case Chomsky's claim that phases of cyclic transfer to the interfaces should correspond with points of access to the lexicon appears somewhat arbitrary. He also observes that given Chomsky's more recent conception as Move as a species of Merge rather than as a composite operation composed of Copy+Merge, it is not clear why Move should be more computationally costly than Merge. As we saw in section 4.2.4, MGBank in fact imposes precisely the opposite Move-over-Merge constraint over derivations.

Chomsky's arguments for phases in terms of the reduction of minimising search between a probe and a goal also appear to be flawed. In Chomsky's structure-driven framework, a probe must search its complement top-down, with phases purportedly limiting the amount of structure to be searched owing to the PIC. However, as noted by Matushansky (2005), the possibility of iterated adjunctions within a phase or of iterated raising constructions, mean that the amount of the structure which must be searched is in principle unbounded. For example, in (244) below, the T probe of the matrix clause must search through three (CP and TP) clauses before it locates its DP goal *several monuments*.

(244) [<sub>CP</sub> There T [<sub>vP</sub> seem [<sub>TP</sub> to have been [<sub>vP</sub> expected [<sub>TP</sub> to be [<sub>vP</sub> constructed several monuments]]]]]]].

Such constructions are the reason Chomsky stipulated that intransitive vP and TP are not phases (recall that the clausal complements of raising and ECM verbs have long been assumed in TG to be defective TPs lacking the CP layer); if unaccusative vP were a phase, then even a simple passive example such as (245) below could not be derived, as the deep object would be trapped by the PIC and therefore would be unable to raise to the subject position, since Chomsky assumes that A-movement, unlike A'-movement, does proceed successive cyclically through phase edges (a point to which we return below).

(245) Jack<sub>i</sub> was [<sub>vP</sub> warned t<sub>i</sub>].

Chomsky's choice of phase heads was therefore intended to allow for A-movement within and across certain clause types. However, as was discussed in sections 3.3.6 and 4.2.4, the MGBank grammar follows Hornstein (2001) in adopting an A-movement

perspective of binding and control. This means that A-movement must now be allowed to cross v\*P and CP phase boundaries, as the following examples illustrate.

(246) Jack<sub>i</sub> wanted [<sub>CP</sub> to t<sub>i</sub> win].

(247) Jack<sub>i</sub> [<sub>v\*P</sub> loves himself<sub>i</sub>].

(248) Jack<sub>i</sub> wanted more than anything [<sub>CP</sub> for himself<sub>i</sub> to win].

In 246, control movement of *Jack* crosses a CP phase boundary unproblematically (infinitival clausal complements of control verbs are standardly assumed to be full CPs); in 247 *Jack* binds across a transitive v\*P phase boundary, and since we are treating the bind of reflexives as an instance of A-movement, this again violates PIC; the same is true of 248 in which the presence of the prepositional complementizer *for* to the left of the subject reflexive indicates that movement for binding has also crossed a phase boundary here, in this case a CP. Once again, then, we see that a decision in one corner of the grammar has repercussions elsewhere: Phase Theory, at least as conceived of by Chomsky, is incompatible with MGBank's analysis of control and reflexive/reciprocal binding.

One possible way around this problem would be to adopt the perspective taken by Boeckx (2009) that in fact both A-movement and A'-movement proceed successive cyclically though every projection between their base position and their final landing site. This *uniform paths* perspective is also adopted for certain types of long-distance dependencies in formalisms which use slash-feature passing mechanisms, such as GPSG and HPSG, or composition such rules, as in CCG. On this perspective, we could say that every XP constitutes a phase and that examples such as 246, 247 and 248 above are grammatical because the A-movers in question have passed through each phase edge.

As noted in (Kobele, 2006, page 41), to a certain extent, MGs already adopt this strongly cyclic perspective on movement: we can view the set of moving chains in any given expression as being situated at the edge of a phase defined by that expression, with the head chain undergoing directly compositional phonetic and semantic interpretation (i.e. being sent to the A-P and C-I interfaces) in lockstep with each application of Merge or Move. Furthermore, as we have seen, the MG presented here relies on a single cycle with overt movements interspersed with covert ones; a separate covert cycle is therefore not required here either. Certain aspects of Phase Theory, in particular the idea of cyclic transfer to the interfaces and the elimination of a separate covert movement cycle, are therefore already an integral part of the MG formalism.



Notice too that the phase-based account of wh-islands is in fact entirely redundant with the intervention-based SMC. Saying that there is no space at the phase edge because there is already a wh-constituent occupying that position is no different in the present framework to the SMC requirement that there be at most one active -wh licensee in the derivation at any one time. Other types of A'-mover, such as topicalised phrases, are also sensitive to wh-islands, of course, as the following example illustrates.

(249) \*That book, I wondered why Pete borrowed.

But as we have seen, these are precisely the sorts of cases which DSMC is designed to block. In fact, Chomsky acknowledges the redundancy between PIC and intervention-based locality constraints when he writes,

*Note that for narrow syntax, probe into an earlier phase will almost always be blocked by intervention effects...It may be, then, that PIC holds only for the mappings to the interface, with the effects for narrow syntax automatic.*

Chomsky (2008, page 10)

It would therefore seem that wh-islands do not provide a strong argument for Chomsky's conception of phases.

With all that said, there does exist some persuasive empirical evidence from both semantics and phonology/morphology that wh constituents (and perhaps all A'-movers) interact with specific intermediate positions (namely CP and vP/VP), rather than just passing passively through them. On the semantic side, we find reconstruction effects such as 250 below (irrelevant traces are omitted throughout this discussion).

- (250) a. \*John<sub>i</sub> thinks that Pete<sub>j</sub> bought [pictures of himself<sub>\*i/j</sub>].  
 b. [Which pictures of himself<sub>i/j</sub>]<sub>k</sub> does John<sub>i</sub> think [<sub>CP</sub> t<sub>k</sub><sup>2</sup> that Pete<sub>j</sub> bought t<sub>k</sub><sup>1</sup>]?

In 250a, the reflexive anaphor *himself* can only be bound by *Pete*, not *Jack*, whereas in 250b, where the DP containing it has undergone wh-movement to the left periphery of the sentence, it can be bound by either. As was discussed in Lebeaux (1985), so-called picture-noun reflexives such as this clearly have different properties from the obligatorily locally bound reflexives which were discussed in section 4.2.4 and which fall under the remit of Chomsky's (1981) Principle A. For example, as 251 below shows, they do not require an sentential antecedent.

(251) Pictures of myself are on display in the gallery.

Hornstein (2001) therefore assumes that picture-NP reflexives are emphatic pronouns or logophors in the sense of Reinhart and Reuland (1993), and therefore are more like pronouns or NOC PRO than true reflexives. As such, they do not constitute residues of A-movement. That being said, picture-NPs do seem to at least strongly prefer taking the closest c-commanding DP as their antecedent, as shown in 250a above. 250b therefore receives a straightforward account if there is an intermediate trace of the *wh* constituent in spec-CP of the embedded clause ( $t_k^2$  in 250b). This would allow the *wh* phrase to reconstruct to this position in which *John* is the nearest c-commander of the reflexive.

Similar reconstruction-based evidence suggests that A'-movement also must also target a position below the surface subject but above the object. Consider the following example from (Fox, 2000, pages 10-11) (cited in (Boeckx, 2009, page 24)).

- (252) [The papers that  $he_i$  wrote for [Mrs Brown] $_j$ ] $_k$ , [every student] $_i$  [ $_{vP}$   $t_k^2$  asked her $_j$  to grade  $t_k^1$ ].

Principle C requires that the referring expression *Mrs Brown* not be bound, while *he* must be c-commanded in order to be bound by *every student*. Accordingly, the topicalised phrase must reconstruct to a position above *her* but below *every student*. The standard assumption is that spec-vP is the position targeted, as indicated by the position of the trace  $t_k^2$  in this example.

Of course, even if spec-vP and spec-CP are indeed targeted by successive cyclic A'-movement, this does not mean that all other projections are not also targeted in the strong sense of actually checking some feature there. As noted by Abels (2003), on the LF side, what we then need is some evidence showing that reconstruction fails when spec-CP and spec-vP are not present. Abels argues that the following paradigm provides such evidence (examples taken from (Boeckx, 2009, page 57)).

- (253) a. \*Mary $_i$  [ $_{vP}$  seems [ $_{TP}$  to John $_j$  [ $_{TP}$  to [ $_{vP}$   $t_i$  like pictures of himself $_j$ ]]]]].  
 b. [Which pictures of himself] $_i$  does it [ $_{vP}$  seem to Mary [ $_{CP}$   $t_i$  that John $_j$  [ $_{vP}$   $t_j$  likes  $t_i$ ]]]?  
 c. [Which pictures of himself] $_i$  does it [ $_{vP}$  seem to John [ $_{CP}$   $t_i$  that Mary $_j$  [ $_{vP}$   $t_j$  likes  $t_i$ ]]]?  
 d. \*Which pictures of himself $_i$  does Mary $_j$  [ $_{vP}$  seem [ $_{TP}$  to John $_i$  [ $_{TP}$  to [ $_{vP}$   $t_j$  like]]]]]?]

253a is ungrammatical because the trace of *Mary* in spec-vP of the *like* clause intervenes between *John* and the reflexive; 253b is fine because the *wh* phrase can reconstruct to its base position from where the reflexive is c-commanded by *John*; 253c is also fine because (by hypothesis) the *wh* phrase containing the reflexive is able to reconstruct to spec-CP of the embedded clause where it can apparently be licensed by *John*.<sup>30</sup> The interesting case is 253d, which features the raising version of *seem* and is quite clearly degraded relative to 253c. On standard assumptions (adopted for MGBank), raising predicates such as *seem* in 253d take bare TP clausal complement lacking the CP layer. Abels therefore argues that the fact that reconstruction is apparently not possible here shows that only CP, not TP, is a target for intermediate *wh* movement (although see Boeckx (2009) for counterarguments).

On the PF side, the evidence that spec-CP is targeted as a landing/feature checking site for intermediate *wh*-movement steps is substantial. One argument comes from the fact that several languages, such as Frisian, Afrikaans, Romani and German allow for a type of *wh*-copying whereby a copy of a fronted *wh*-word appears in every intermediate spec-CP position through which it (by hypothesis) moves. The following example from German is from Felser (2004).

- (254) Wen glaubst du, wen Peter meint, wen Susi heiratet? (German)  
 Who believe you who Peter thinks who Susi marries?  
 ‘Who do you believe Peter thinks that Susi is marrying?’

In the context of Chomsky’s copy theory of movement, the presence of the intermediate versions of *who* can be explained as an instance of lower copy spell out. Kobele (2006) shows how a phonetic copying mechanism can be incorporated into MGs, albeit at the cost of raising their expressive power to that of Parallel Multiple Context Free Grammars (equivalent to Range Concatenation Grammars - still polynomial but not mildly context sensitive). This copying mechanism could be used to generate these examples.

A further piece of evidence for successive cyclic *wh*-movement through spec-CP comes from Irish. Like English, this language has a declarative complementizer *go* (‘that’) (255a). However, in clauses with a fronted *wh*-word, we find instead the

<sup>30</sup>A complicating factor is that *John* appears not to c-command the reflexive given that he is embedded inside a PP. This is not of immediate concern for us, however, given that MGBank does not handle the co-indexing of picture-NP reflexives in the syntax anyway (but see Pesetsky (1996), Kitahara (1997) and Boeckx (1999) (all cited in Boeckx (2009)) for discussion of how to obtain the required c-command configuration).

particle *aL* in the complementizer position (255b), and where wh-movement is long-distance, we find that all intermediate CPs are headed by this particle (255c).

- (255) a. Creidim gu-r inis sé bréag  
believe-1-SG go-PAST tell he lie  
'I can't believe that he told a lie.'
- b. Céacu ceann a dhíol tú  
which one aL sold you  
'Which one did you sell?'
- c. an t-ainm a hinnseadh dúinn a bhít ar an áit  
the name aL was.told to.us aL was on the place  
'The name that we were told was on the place'.

Given that in Minimalism movement is argued to be driven by the need to check and delete certain morphosyntactic features, the morphosyntactic marking of the intermediate complementizer here suggests that the null wh-operator, which is by hypothesis positioned in the matrix spec-CP, has also passed through the intermediate spec-CP position where it checked some feature.

There is also good evidence on both the LF and PF side that wh constituents must proceed successive cyclically through (at least) transitive verb phrases, although here the evidence does not seem to me to distinguish between spec-VP and spec-vP as possible landing sites. For example, we again find evidence from reconstruction.

- (256) [Which picture of himself]<sub>i</sub> does [<sub>TP</sub> Pete<sub>j</sub> [<sub>vP</sub> t<sub>j</sub> [<sub>v'</sub> t<sub>i</sub> expect Mary<sub>k</sub> [<sub>TP</sub> to t<sub>k</sub> bring t<sub>i</sub>]]]]?

In this example, *Pete* is able to bind the reflexive, suggesting that the wh phrase containing that reflexive reconstructs to some position above *Mary* (given the closest c-commanding antecedent restriction on picture-NP reflexives) but below *Pete*. This position could be either spec-VP or spec-vP, but here we will assume it is the latter in line with the standard assumptions of MP.<sup>31</sup>

<sup>31</sup>Notice that if wh movement proceeds through spec-vP, then 250b no longer constitutes evidence of successive cyclic movement through spec-CP, as now the wh phrase containing the reflexive will stop in spec-vP of the *think* clause from where it can be bound by *John* from spec-TP. However, in 1b below, *John* appears in spec-VP, below the spec-vP position, and is able to bind the reflexive, suggesting that the wh phrase has indeed passed through spec-CP.

- (1) a. \*Mary told Jack that Susan likes that picture of himself.  
b. [Which picture of himself]<sub>i</sub> did Mary [<sub>vP</sub> t<sub>i</sub> [<sub>v'</sub> tell<sub>j</sub> [<sub>VP</sub> Jack<sub>i</sub> [<sub>V'</sub> t<sub>j</sub> [<sub>CP</sub> t<sub>i</sub> [<sub>C'</sub> that Susan [<sub>vP</sub> t<sub>i</sub> [<sub>v'</sub> likes t<sub>i</sub>]]]]]]]]?

One piece of evidence from morphosyntax for intermediate wh-movement through the verb phrase comes from French, where the past participle form of a verb which takes *avoir* ('have') as its perfect auxiliary must agree in gender and number with its direct object just in case that direct object precedes it. This is illustrated by the following examples.

- (257) a. Tu a mis(\*es) quelle lettres sur mon bureau?  
 You have put-MASC.SG/\*FEM.SG which letters on my desk?
- b. Quelle lettres a tu mis(es)\* sur mon bureau?  
 Which letters have you put-FEM.PL/\*MASC.PL on my desk

In 257a the feminine plural direct object *quelle lettres* appears to the right of the verb which obligatorily appears in its default masculine singular form. In 257b, on the other hand, the direct object appears in the left periphery, and the past participle form of the verb must now appear in its feminine plural form. One explanation for these facts is that this agreement takes place locally between the verb and the wh constituent when the latter moves through the verb phrase on its way to the left periphery.<sup>32</sup>

A further piece of morphological evidence for successive cyclic wh movement through the verb phrase comes from the following Bahasa Indonesia data presented in Saddy (1991) (cited in Hornstein et al. (2005, page 361)).

- (258) a. Bill men-gira Tom men-harap Fred men-cintai siapa?  
 Bill TR-thinks Tom TR-expects Fred TR-loves who?
- b. Siapa yang Bill Ø-kira Tom Ø-harap Fred Ø-cintai?  
 who FOC Bill think Tom expect Fred love  
 'Who did Bill think (that) Tom expects (that) Fred loves?'

Example 258a exemplifies a wh in situ construction in this language while 258b is the corresponding wh movement sentence. The relevant point of note here is that the transitive prefix *men* which appears on all the verbs in 258a has been dropped in 258b. This is plausibly interpreted as a morphological reflex of successive cyclic wh movement from each intermediate verb phrase on the way to the matrix spec-CP position.

The evidence for successive cyclic movement through CP and vP/VP is, to my mind, compelling (for a much more extensive survey of this evidence, see Radford (2004) and Hornstein et al. (2005)). Furthermore, the morphosyntactic marking of

<sup>32</sup>Note that if we assume wh in situ constructions to involve covert movement of the wh item to spec-CP, then we would have to refine this argument to the effect that only overt movement is 'strong' enough to trigger the agreement effect on the verb here.

certain intermediate landing sites suggests that this movement is not merely passive but actually involves some kind of feature checking at these specific positions. For this reason, MGbank includes explicit intermediate movement steps via spec-CP and spec of transitive vP (Chomsky's v\*P). At present, this is implemented for wh-movement only.

How to actually implement intermediate A'-movement steps has proven to be something of a vexed question in TG. Take the case of successive cyclic movement of an interrogative wh constituent through one or more intermediate CPs and vPs. The problem is that the movement is clearly motivated by the need for the wh constituent to enter into a checking relation with the final C head in whose spec it will land, not with any of the intermediate C and v heads. It is therefore unclear how or why such intermediate movement takes place. One answer to the why question is that constituents must move to the edge of each vP and CP phase in order to escape the freezing effects of the PIC. We have already seen that Chomsky's specific proposals for phases are not compatible with the framework proposed here.

It is beyond the scope of this thesis to attempt a comprehensive answer to the *why* question. One (highly speculative) possibility is that the intermediate movement steps, at least via spec-CPs, may derive from some kind of analogical process within the lexicon which 'weakly' spreads whatever feature triggers final wh-movement to interrogative C heads to other C heads. It has also been argued that in certain languages, overt wh-movement targets either spec-vP or some position just above it as a final landing site (see, e.g., Aldridge (2009) on Old Japanese and Manetta (2010) on Hindi). This suggests that there may be a low focus<sup>33</sup> head in the vP domain which could be the target of intermediate wh movement; in languages without such 'short' wh movement, this otherwise dormant focus head may retain some sort of weak wh licenser which is strong enough to attract an already moving wh constituent to its specifier.

What is highly relevant if we wish to implement successive cyclic wh movement here is the *how* question. Here, Rizzi (2006) characterises the problem in the following terms.

*The paradox of these intermediate positions is that, on the one hand they must autonomously cause a movement step (if we want to take seriously the idea that each step is locally determined, with no "look-ahead" to subsequent derivational steps), and at the same time we should make sure that movement should not stop there.*

---

<sup>33</sup>Interrogative wh movement is often regarded as a type of focus movement - see, e.g., Rizzi (1997).

(Rizzi, 2006, page 9)

Here we will implement successive cyclic wh movement by assuming that in addition to the ! type deleting suicidal licensors which were introduced in section 3.3.4, there are also non-deleting suicidal licensors which are marked with ?, an example of which is +wh?. We will add these features to all complementizer heads and the [trans] little v head. For example, the new categories of complementizer *that* and transitive little v (simplified by omitting the fine-grained selectional features) are given below.

*that* :: t= +wh? c

[trans] :: >v= +self! +wh? =d lv

Non-deleting suicidal licensors have the following properties.

- (259)
1. If the active feature of the head chain of an expression is a +x?/+X? suicidal licensor and the derivation does not contain a moving chain with an active -x feature, then +x? simply self-deletes, allowing the derivation to continue.
  2. If there is a -x feature, but the subcategorization requirements (e.g. +NOM) of the +x? licensor are not met, the derivation aborts.
  3. If there is a -x feature and the subcategorization requirements of the +x? licensor are met, the +x? licensor is deleted but the -x licensee persists.
  4. A ?-type suicidal licensor may be an overt licensor (e.g. +WH?) or a covert licensor (+wh?), but in either case, if this licensor is checked by an overt mover, that mover must continue moving overtly.

The two differences between ! and ? type suicidal licensors are 1. that the ? type fails to delete the licensee feature it checks (because we want the mover to keep moving), and 2. that ? type licensors do not allow a mover to move overtly to check them and then to transition to being a covert mover. This last condition ensures that we avoid generating examples such as 260 below, which is precisely the scenario Rizzi was referring to.

(260) [<sub>CP</sub> μ<sub>i</sub> Does Jack [<sub>vP</sub> μ<sub>i</sub> believe [<sub>CP</sub> who<sub>i</sub> Pete [<sub>vP</sub> t<sub>i</sub> likes t<sub>i</sub>]]]]

(With the meaning: *who does Jack believe Pete likes?*)

In this example, *who* moves overtly to the intermediate spec-CP, and then carries on covertly to its final landing site (both the final and intermediate covert landing sites are marked by  $\mu_i$ ), meaning that the overt copy is spelled out in the wrong spec-CP. We could of course avoid this scenario by assuming that the final interrogative C head's licenser is necessarily an overt +WH. This would force *who* to undergo overt movement the whole way. Unfortunately, MGBank also includes an interrogative C head whose wh licenser is a covert +wh in order to capture echo questions such as 261 below in which the wh item remains in situ.

(261) Jack said that Pete likes *who*!?

Condition 4 of 259 above is therefore required by the MGBank grammar in order to block examples such as 260. At an explanatory level, the reason +wh? features cannot support overt constituents in spec of the head bearing them is perhaps related the fact that, unlike strong +WH (and +wh) licensers, they are not associated with any semantic interrogative value/feature in English and other languages without short wh movement, hence their inherent weakness relative to their strong counterparts.

Formally, we can encode these various properties of ?-type suicidal licensers using the rules given in fig 4.15, where +f!? can be either an overt or covert, !- or ?-type suicidal licenser, and both  $\gamma$  and  $\delta$  are feature sequence suffixes which may be empty.

$$\frac{[s: +f!? \gamma, \alpha_1, \dots, \alpha_{i-1}, t: \delta, \alpha_{i+1}, \dots, \alpha_k]}{[s: \gamma, \alpha_1, \dots, \alpha_{i-1}, t: \delta, \alpha_{i+1}, \dots, \alpha_k]} (fcide)$$

$$\frac{[s: +f? \gamma, \alpha_1, \dots, \alpha_{i-1}, t: -f \delta, \alpha_{i+1}, \dots, \alpha_k]}{[s: \gamma, \alpha_1, \dots, \alpha_{i-1}, t: -f \delta, \alpha_{i+1}, \dots, \alpha_k]} (smove)$$

Figure 4.15: Sub-functions of MOVE

We already saw the rule *fcide* in section 3.3.4 because it applies to both ! and ?-type suicidal licensers. As was noted in that earlier section, this rule does not involve any movement; instead, the ? simply checks and deletes itself because no matching licensee is present. Unlike with !-type licensers, the other Move rules presented in this thesis do not apply to ?-type licensers. Instead, these features are used in just one movement rule *smove* shown in fig 4.15. In this rule, a -f licensee is present which matches the +f? licenser and which furthermore satisfies all fine-grained selectional properties. As a result, movement applies but does not concatenate the string of the



mover with the string of the head owing to the fact that only the licenser is deleted, not the licensee. And because like the other Move rules, the rule *p\_move* from section 3.3.4 (which concatenated the string of the mover with the string of the head chain and then allowed the mover to keep moving covertly) does not apply to *?-type* licensers, this ensures that examples such as 260 cannot be generated.

With our *+wh?* licensers distributed across all intermediate C and (transitive) little *v* heads, *wh* movement will proceed successive cyclically through intermediate (transitive) *spec-vP* and intermediate CP heads either covertly (in the case of echo questions) or overtly (in non-echo questions).<sup>34</sup> This enhances the cross-linguistic descriptive adequacy of the grammar, enabling a straightforward account of the French data in 257b, for example. The MGBank gramma would represent this sentence as in fig 4.16, where the *wh*-phrase *quelle lettres* moves successive cyclically through the inner *spec-vP* position on its way to *spec-CP* in the left periphery. This sets up the opportunity to enforce the gender and number agreement on the verb.

In derivational terms we can enforce such agreement if we assume the following two categories in French for the null transitive light verb and the past participle verb in its feminine plural form.

[trans] :: >v{x}= +self! +wh{x}? =d lv{x}  
 mises :: d= +CASE{+ACC} v{+FEM.+3PL.PERF}

The selectional variable on the *>v=* selector and *+wh?* licenser of the [trans] feature will ensure that the *+FEM* and *+3PL* requirements on the past participle percolate onto this *+wh?* licenser where they will enforce the necessary feminine 3rd plural agreement on the object as it passes through *spec-vP*. Such percolation would also take place in the case of example 257a, but in this case the direct object never passes through *spec-vP*, hence the agreement will not be enforced and the sentence is predicted to be grammatical. Note that we would also need to use a version of the perfective auxiliary which does not percolate its complement's selectional features; otherwise the object agreement features would percolate up to the *+CASE* of T alongside the subject agreement features, and the 2nd person subject would fail to satisfy the *+3PL* and *+FEM* requirements causing the derivation to crash (though this could also be avoided by using disjunctive features such as *+FEM|NOM* and *+3PL|NOM* to enforce the object agreement).

<sup>34</sup>Recall from section 3.3.4 that covert licensers (including suicidal licensers) can be checked via overt movement provided that the mover continues moving and eventually checks some overt licenser.

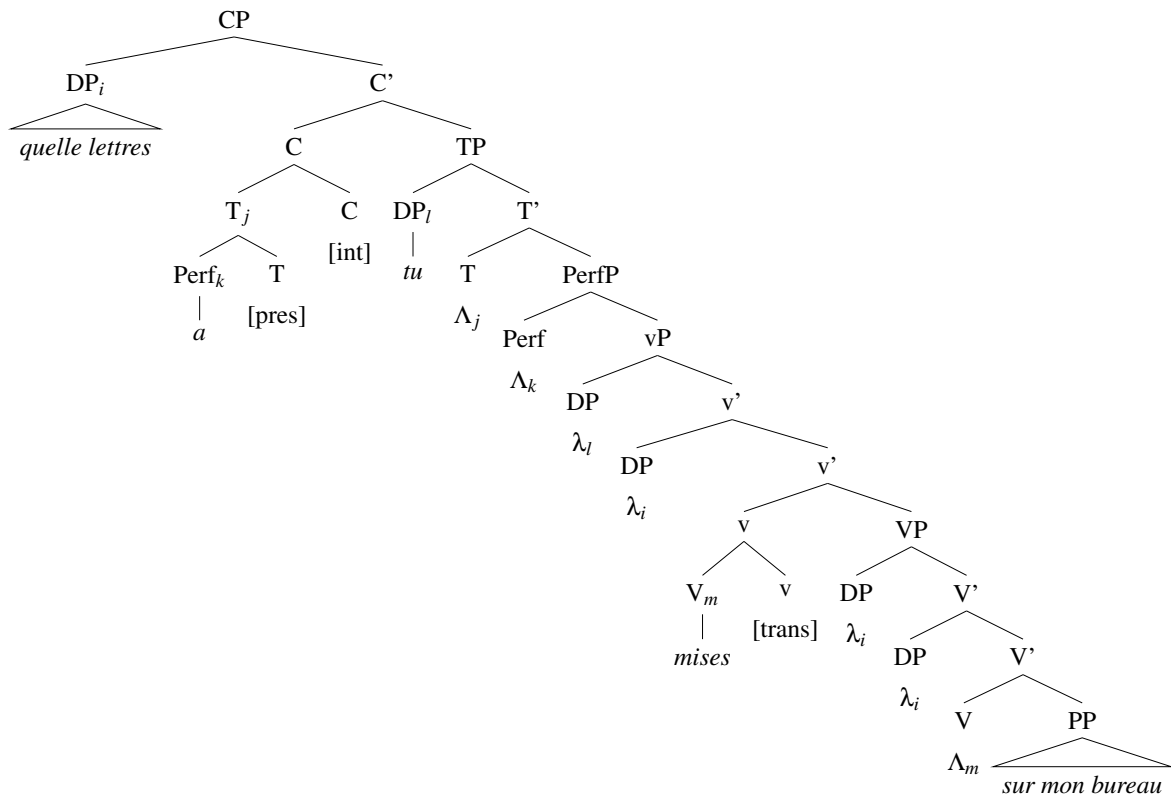


Figure 4.16: Derived Xbar tree for the French sentence *Quelle lettres a tu mis(es)\* sur mon bureau?* ('which letters have you put on my desk?') which features agreement between a past participle verb and its preceding direct object and constitutes evidence for successive cyclic wh-movement through the verb phrase.

The approach to successive cyclic A'-movement adopted here could similarly be used to provide accounts of the other phenomena discussed in support of this phenomenon in the section (the derivations are left as an exercise). However, the wh-copying phenomena exemplified for German in 254 would also require the addition of copying operations of the type proposed in Kobele (2006), which would increase the expressive power of the formalism to that of PMCFG.

### 4.3.5 The Complex NP Constraint

Another example of a containment-based locality constraint (discovered by Ross (1967)) is the so-called Complex NP Constraint (CNPC), illustrated below.

- (262) a. Who<sub>i</sub> does Jack believe [<sub>CP</sub> (that) Mary likes t<sub>i</sub>]?  
 b. Jack believes the [<sub>NP</sub> claim (that) Mary likes *who*]?  
 c. \*Who<sub>i</sub> does Jack believe the [<sub>NP</sub> claim (that) Mary likes t<sub>i</sub>]?

In 262a, *who* is extracted out of an embedded clause and the sentence is fine. In 262b the *like* clause is embedded inside an NP and the wh item remains in situ, resulting in an acceptable echo question. In 262c, on the other hand, the wh item moves across the NP boundary and this results in an unacceptable sentence.

There have been attempts to explain the CNPC by appealing to (a modified version of) Phase Theory and the PIC (see e.g. Bošković (2015)). We saw in the previous section that this approach to locality is incompatible with certain aspects of the MGBank grammar (in particular its treatment of control and reflexive/reciprocal binding as involving A-movement). MGBank therefore simply stipulates the effects of the complex NP constraint in the lexicon using suicidal licensors for every type of A'-movement, most of which have dummy +NONE selectional requirements. For example, the category for the noun *claim* when taking a clausal complement is given (in simplified form) below.

*claim* :: c= +wh{+ECHO}? +foc{+NONE}! +top{+NONE}! n

Because no lexical item in the MGBank lexicon has a NONE property on any of its features, the above category effectively prevents focus or topic movement from crossing the NP barrier. The +wh{+ECHO}? licensor prevents any wh item from crossing it except one that is moving to form an echo question. Recall from section 3.3.4 that echo questions are treated as involving covert wh movement in MGBank, rather than

genuine wh-in situ (this is so that we can capture the meaning of wh interrogatives uniformly for both echo and non-echo wh questions, although the echo variety clearly involve an additional emphatic focus layer of meaning). Using +ECHO instead of +NONE here therefore allows MGBank to generate examples such as 262b, while still blocking examples such as 262c. The choice between ! and ? type suicidal licensors for the licensors with the dummy +NONE requirements is arbitrary here: both would have the same effect. We need the +wh{+ECHO}? licensor to be a ?-type licensor, however, because it must not delete the -wh licensee which is moving covertly through this position on its way to some higher spec-CP final landing site.

### 4.3.6 Interrogative clause islands

We have already seen that SMC blocks wh-island violations in MGBank. However, even interrogative clauses without fronted wh-items often constitute islands, as illustrated in 263 below.

(263) \*Who<sub>i</sub> did Jack ask if/whether Mary likes t<sub>i</sub>?

MGBank treats both *if* and *whether* as interrogative complementizers,<sup>35</sup> hence neither has a -wh feature to check and so SMC will not block 263.

Interrogative clause island violations of this kind are blocked in MGBank by assigning to *if/whether* the following category.

$$claim :: t = +wh\{+ECHO\}! +foc\{+NONE\} +top\{+NONE\} c$$

This strategy is very similar to that adopted for complex NP islands in the previous section, except that here the +wh! licensor is of the ! type because for echo questions such as 264 below the embedded spec-CP is necessarily the final landing site of the covertly moved wh item.

(264) Jack asked [<sub>CP</sub> μ<sub>i</sub> whether/if Mary likes *who*<sub>i</sub>]?!]

<sup>35</sup>An alternative approach pursued for *whether* is to treat it as some kind of adverbial wh-operator, which checks the wh licensor of a null interrogative C head. In Radford (2004), for example, it is suggested that *whether* is merged directly into spec-CP to check the wh (and epp) feature of interrogative C. This approach is incompatible here, however, as the +WH licensor of interrogative C must be checked by Internal Merge (Move) not External Merge. We would therefore need to assume that *whether* starts out adjoined to some projection below C, before moving to spec-CP to check +WH. Instead of pursuing this non-standard strategy, MGBank simply treats *whether* as a complementizer.

## 4.4 Subject/non-subject asymmetries

In this section we look at how the MGbank grammar enforces certain curious asymmetries between subject and non-subject wh-movers.

### 4.4.1 Complementizer-trace effects

#### 4.4.1.1 *that*-trace effects and property suppressor features

The first asymmetry which we will consider is the so-called *that*-trace effect first identified by Perlmutter (1968), who noted that in many languages, including English, an overt declarative complementizer such as *that* cannot head a clause whose subject has been A'-extracted. This is illustrated for wh-movement below.

(265) Who<sub>i</sub> did Jack say (that) Mary helped t<sub>i</sub>?

(266) Who<sub>i</sub> did Jack say (\*that) t<sub>i</sub> helped Mary?

As Pesetsky (2015) observes, many different approaches to this phenomenon have been proposed, and yet still no consensus exists on which is correct. The fact that it is found in so many diverse and unrelated languages suggests a deep explanation rooted in fundamental principles of language, which should manifest themselves in other areas of the grammar. However, it is not our intention to attempt such an explanatorily adequate<sup>36</sup> account here. Instead, our more modest and practical goal is to enhance the descriptive adequacy of the grammar and in so doing to reduce the parser's search space. In this section we will look at how *that*-trace effects have been enforced for wh-movement in MGbank.<sup>37</sup>

In order to enforce complementizer-trace effects, the grammar needs to know whether a given wh-word is functioning as a subject or a non-subject. In the case of finite clauses, we could potentially exploit the fact that the subjects of these clauses must be nominative. However, the MGbank formalism currently does not include a mechanism for feature valuation;<sup>38</sup> instead, as in Chomsky (1995), all items enter the derivation fully specified for case. However, as we saw in section 3.4.1, the MGbank treats items

<sup>36</sup>See Chomsky (1965) on the levels of adequacy (observational, descriptive, explanatory) of linguistic theories.

<sup>37</sup>At present, these effects are not enforced for other types of A'-movement, but this could be achieved with relatively minor alterations to the grammar (for example, by splitting [topicalizer] heads into nominative and non-nominative versions).

<sup>38</sup>In Torr and Stabler (2016) such a mechanism was introduced but this has since been removed in order to remove the redundancy of having two mechanisms by which morphosyntactic case is specified.

such as *you, it, those* etc, which are syncretised for case in English, as being specified with both NOM and ACC (and in some cases also GEN) case.

In order to capture *that*-trace effects here, we will therefore need to sacrifice some succinctness in the lexicon by splitting wh-words such as *which* and *who*, which are also syncretised for case in English, into separate nominative and accusative versions, as shown below.

$$who :: d -case\{3SG.NOM.WH\} -wh\{NOM\}$$

$$who :: d -case\{3SG.ACC.WH\} -wh\{ACC\}$$

The first *who* bears a NOM property meaning it can only be case-licensed in spec of finite TP (i.e. as the subject of a finite clause), while the second version is marked as ACC and so must be licensed as the object of a verb or preposition. Note also that the -wh licensee effectively records a history of this case licensing position because it is marked with the same case property as the -case licensee preceding it. For sentences with just one level of clausal embedding, we can now enforce *that*-trace effects using the following categories for the overt and null declarative complementizers respectively.

$$that :: t\{+FIN.x\} = +wh\{-NOM\} ? c\{DECL.THAT.x\}$$

$$[decl] :: t\{x\} = +wh ? c\{DECL.SUBORD.x\}$$

The *that*-trace effect is enforced by the -NOM selectional requirement on the ? type suicidal wh licenser of *that*, which as we saw in section 4.3.4 is used to enforce successive-cyclic wh movement in MGBank. As was noted in that earlier section, although suicidal features will self-delete if no matching licensee is found, where one is present, that licensee must satisfy any selectional restrictions located on the licenser, or the derivation will abort. Thus the -NOM requirement ensures that whenever the overt finite complementizer appears, extraction of a nominative wh-word across it is impossible, thus correctly enforcing the *that*-trace effect in 266. The null [decl] C head, meanwhile, contains no such restriction, and thus will correctly allow a nominative wh word to be extracted across it.

Unfortunately, while this approach is sufficient for cases where wh-extraction crosses a single CP boundary, it currently undergenerates with respect to the following perfectly acceptable example.

(267) [<sub>CP</sub> Who<sub>i</sub> did he [<sub>vP</sub> t<sub>i</sub><sup>6</sup> say [<sub>CP</sub> t<sub>i</sub><sup>5</sup> that she [<sub>vP</sub> t<sub>i</sub><sup>4</sup> thinks [<sub>CP</sub> t<sub>i</sub><sup>3</sup> [<sub>TP</sub> t<sub>i</sub><sup>2</sup> [<sub>vP</sub> t<sub>i</sub><sup>1</sup> helped him?]]]]]]]]

The problem here is that wh-movement of nominative *who* crosses two CPs, the second of which is headed by an overt *that* complementizer whose -NOM requirement will at present incorrectly block this movement. Clearly the subject/non-subject asymmetry with respect to wh-extraction across *that* only applies within the clause in which the nominative subject is case licensed; once this subject has exited that case-licensing clause, it may freely cross any instances of *that* located in higher clauses.

To resolve this issue, we now introduce a new type of *property suppressor* feature into the inventory of fine-grained selectional features. These features have the form  $\sim X$ , where  $X$  is some selectional property. Including a selectional suppressor on a licenser (or selector) has two effects: 1. it prevents the specified property from undergoing percolation; 2. the property is deleted from the checked licensee or selectee, which is relevant if that feature persists and causes the chain containing it to begin or continue moving. For example, we can use a  $\sim$ NOM suppressor to knock out the NOM property on *who*'s -wh licensee *after* it leaves the case-licensing CP clause *but before* it arrives at the next spec CP position, thereby preserving the *that*-trace effect in sentences like 266 while eliminating it from sentences like 267. Recall that following standard assumptions in MP, wh movement proceeds successive-cyclically in MGBank through all intermediate transitive spec CP *and* spec vP positions. We can therefore achieve the desired result by placing the  $\sim$ NOM suppressor feature on transitive little  $v$ 's +wh? suicidal licenser, as shown below:

$$[\text{trans}] :: >v\{x\} = +wh\{\sim\text{NOM}\}? =d \text{lv}\{x\}$$

After being case licensed at  $t_i^2$  in 3, *who* moves (via  $t_i^3$ ) to  $t_i^4$ , where it has the NOM feature on its -wh licensee deleted by the  $+wh\{\sim\text{NOM}\}?$  licenser. It then moves to  $t_i^5$  where it now satisfies *that*'s [+REL|-NOM] requirement, and finally proceeds (via  $t_i^6$ ) to the matrix spec CP position.

#### 4.4.1.2 anti-*that*-trace effects

Interestingly, while complement clauses head by *that* do not allow extraction of their subjects, for relative clauses lacking any overt wh-operator, we find the reverse situation: if the constituent relativized is the subject, *that* is obligatory (268), while for non-subjects the presence of *that* is again optional (269).

(268) The man<sub>*i*</sub> \*(that) *t<sub>i</sub>* helped Mary..

(269) The man<sub>*i*</sub> (that) Mary helped *t<sub>i</sub>*..

In this section we will look at how anti-*that*-trace effects are enforced in MGBank. To begin, note that when *that* appears in the left periphery of a restrictive relative clause, it is standardly treated as a complementizer rather than as a relative pronoun in Minimalism. Radford (2004, pages 228-229) summarises some of the evidence for this view, which includes the fact that relative *that* has the same unreduced (/ðæt/) and reduced (/ðət/) phonetic exponents as complementizer *that* in contrast to the determiner/pronominal *that* which only has the unreduced form, the fact that it cannot pied-pipe a preposition in contrast to wh relatives such as *who(m)* (270) and the fact that it has no genitive form, again in contrast to wh relatives (271).

(270) the man to \*that/who(m) she sent the letter

(271) the man \*that's/whose car is being serviced

MGBank treats *that* in relative and complement clause contexts as one and the same item (which is not, of course, necessary even if one adopts the position that relative *that* is indeed a complementizer). The details of the MGBank analysis of restrictive relative clauses with an overt wh operator were given in section 4.2.1, where it was noted that for relative clauses lacking an overt wh-operator (including *that* relatives), MGBank adopts the standard Minimalist assumption that a null [wh] wh-operator is nevertheless present, with this item moving to spec CP to check +WH/-wh features much as its overt counterparts do. It was also noted in that section that the only difference between overt and covert wh operators in MGBank is that in the latter case only, the wh operator remains attached to the relativized NP so that it can undergo overt movement to check the overt licensor +WH (recall that movers lacking phonetic content are regarded here as only being capable of undergoing covert movement). This is achieved using the familiar ‘pied-piping’ simulation mechanism, with the [wh] operator deleting the -n feature of its relativized NP complement, replacing it with its own. In order to enforce the anti-*that*-trace effects, we will again need to distinguish nominative from non-nominative wh-movers. To do this, we will use the following two categories of null [wh] operator used for nominative and accusative relativized constituents respectively (a special [wh] operator is used in MGBank for the case where the relativized element is an adverbial (e.g. *the reason [wh] (that) he did it..*), which we omit from the discussion here).

[wh] :: n= +N D -case{NOM} -wh{NOM.OP} -n

[wh] :: n= +N D -case{ACC} -wh{OP} -n



These two null heads would be used to generate examples 268 and 269 above respectively. To enforce the anti-*that*-trace effect in 268, we will split the [rel] C head into two versions (recall from section 4.2.1 that the [rel] head bears the +WH licenser that attracts the wh constituent in relative clauses), which are given below (as usual, in simplified form).

$$\begin{aligned} [\text{rel}] &:: c\{+\text{THAT}\}=\text{+WH}\{+\text{OP}\} c\{\text{RELAT}\} \\ [\text{rel}] &:: c\{-\text{THAT.}+\text{DECL.}-\text{INF}\}=\text{+WH}\{-[\text{NOM}—\text{OP}]\} c\{\text{RELAT}\} \end{aligned}$$

In addition, we amend our earlier category for complementizer *that* as follows so that it includes the selectional property THAT.

$$\textit{that} :: t=\text{+wh?} c\{\text{THAT}\}$$

The first [rel] head must select a CP headed by *that* owing to the +THAT selectional requirement on its c= selector. It will then attract a wh constituent which has an OP selectional property on its -wh licensee. Only the null [wh] operators are marked with the feature OP, not overt wh operators (*which*, *who* etc). This enforces the constraint that an overt wh item cannot co-occur with *that*, which is one of the effects of the so-called Multiply Filled COMP Filter proposed by Chomsky and Lasnik (1977) - see section 4.7.6). This will block examples such as 272 below.

(272) \*The man who that Mary helped..

This first [rel] head does not restrict the case of its wh attractee, hence both nominative and accusative wh movers are correctly licensed in the presence of *that*. The second [rel] head selects for a declarative non-infinitival CP which is not headed by the overt complementizer *that*. The only C head satisfying this is the null [decl] C head. This [rel] head then attracts a wh mover which is either overt *or* non-nominative, which it does using the disjunctive selectional requirement feature -[NOM—OP]. This means that for relative clauses lacking *that*, the wh item in the left periphery must either be overt or non-nominative, which correctly enforces the anti-*that*-trace effect seen in 268 while also allowing for examples such as 273 below, where the nominative relativized item is permitted owing to the presence of the overt wh word.

(273) The man who helped Mary..

The reader may have noticed that there is now a conflict in our grammar between the competing demands of the *that*-trace and anti-*that*-trace effects, and that in actual

fact our grammar currently undergenerates the simple example in 274a below, with the structure in 274b.

- (274) a. The man that helped Mary is here.  
 b. The  $[_{NP} [[wh] man]_i$  [ $nom$ ]  $[_{CP} t_i$  [ $rel$ ]  $[_{CP}$  that  $t_i$  helped Mary is here]]].

As we saw in the previous section, the +wh? suicidal licenser on *that* bears a -NOM feature which enforces the *that*-trace effect. The problem here is that, as indicated in 274b, the nominative wh mover must be allowed to cross *that* on its way to check the +WH licenser of the [ $rel$ ] head. In order to allow for this, we will further split overt wh words into separate relative and non-relative versions, marking the -wh licensee of the former with the feature RELAT. This feature will also appear on null [wh] operators. The categories for the relative and non-relative versions of nominative *who*, together with the updated [wh] heads are given below.

$who :: n\{+REL\}= d -case\{3SG.NOM.WH\} -wh\{NOM.RELAT\}$   
 $who :: d -case\{3SG.NOM.WH\} -wh\{NOM\}$   
 $[wh] :: n= +N D -case\{NOM\} -wh\{NOM.OP.RELAT\} -n$   
 $[wh] :: n= +N D -case\{ACC\} -wh\{OP.RELAT\} -n$

We then replace the simple -NOM feature on the +wh? licenser of *that* with the disjunctive feature [-NOM—+RELAT].

$that :: t\{+FIN.x\}= +wh\{[-NOM—+RELAT]\} ? c\{DECL.THAT.x\}$

The result of this is that only interrogative nominative wh movers will be blocked by *that*, not relativized ones, which allows us to generate example 274a. One problem for this approach is that we now overgenerate examples such as 275 below, in which the relativized wh item moves to spec-CP of a higher clause.

- (275) \*An amendment which they say that will be law next year..

One way around this problem would be to use separate versions of *that* for relative and non-relative clauses. Non-relative *that* would then use the simple -NOM requirement instead of the disjunctive [-NOM|+RELAT] to enforce the *that*-trace effect, while relative *that* would lack this feature. This is not implemented in MGBank at present, however.

#### 4.4.1.3 *for*-trace effects

Complementizer-trace effects are also found in infinitival clauses headed by the prepositional-complementizer *for*.

(276) Who<sub>i</sub> would you prefer (for) him to work with t<sub>i</sub>?

(277) Who<sub>i</sub> would you prefer (\*for) t<sub>i</sub> to work with him?

Because the wh-subject in 277 is accusative not nominative, we cannot use a -NOM feature to enforce the *for*-trace effect as we did for the *that*-trace effect. Notice, however, that the accusative entry for *who* above included a WH property on its -case licensee. The MGBank analysis of *for-to* constructions involves a derivational step in which *for* checks the accusative case of the infinitival clause's subject (see section 4.7.6). MGBank therefore blocks *for*-trace configurations (for wh-movement) simply by including a -WH negative selectional requirement on *for*'s +CASE licensor.<sup>39</sup>

#### 4.4.2 *Do*-trace effects

Another example of a subject/non-subject asymmetry is the so-called *do*-trace effect. English main clause wh interrogatives require the presence of an auxiliary, and where no semantically meaningful auxiliary is present the dummy auxiliary *do* must be inserted. The one exception to this is where the questioned wh constituent is the subject of the interrogative clause itself, in which case an auxiliary is not obligatory and (non-emphatic) *do* cannot be inserted.

(278) [Which people]<sub>i</sub> \*(did) Jack see t<sub>i</sub> every day?

(279) [Which people]<sub>i</sub> (\*did) t<sub>i</sub> see Jack every day?

Where the extracted wh-item is the nominative subject of an embedded clause, however, *do* is still required.

(280) [Which people]<sub>i</sub> \*(did) Mary say t<sub>i</sub> saw Jack every day?

In other words, the *do*-trace effect only applies within the clause in which the nominative wh mover is case licensed, as was the case for the *that*-trace effect which we saw in section 4.4.1.1.

<sup>39</sup>In actual fact, the disjunctive feature [+ECHO|-WH] is used rather than -WH in order to allow for echo questions like *you would prefer for WHO to work with him?!*

The MGBank grammar enforces *do*-support and the *do*-trace effect by using two separate interrogative C heads for subject and non-subject main clause wh-questions. The (simplified) categories for these are given below.

$$\begin{aligned} [\text{int}] &:: >t\{+\text{FIN}.[+\text{EMPH}|\text{-DO}].x\}= +\text{WH}\{+\text{NOM}\} c\{\text{INT.MAIN}.x\} \\ [\text{int}] &:: >t\{+\text{FIN}.\text{+AUX}.x\}= +\text{WH}\{\text{-NOM}\} c\{\text{INT.MAIN}.x\} \end{aligned}$$

The disjunctive selectional requirement  $[+\text{EMPH}|\text{-DO}]$  on the  $>t=$  selector of the first C head ensures that non-emphatic *do* cannot head its TP complement (other auxiliaries are permitted but not obligatory), while the  $+\text{NOM}$  selectional requirement on the  $+\text{WH}$  licenser ensures that the wh-constituent will be the subject of a finite clause. The  $+\text{AUX}$  on the  $>t=$  selector of the non-subject C head, meanwhile, specifies that the TP complement must be headed by an overt auxiliary (which may or may not be dummy *do*) which will undergo T-to-C head movement when C is merged with TP; the  $\text{-NOM}$  on the  $+\text{WH}$  then ensures that the raised item cannot be a finite clause subject. Examples such as 280 above are still generated though, because the same mechanism which negates the *that*-trace effect for nominative wh-movers which have exited their case licensing clause will also negate the *do*-trace effect in this situation: the  $[\text{trans}]$  little *v* head of the embedding clause has a  $\sim\text{NOM}$  selectional suppressor feature which will eliminate the  $\text{NOM}$  property on the moving wh constituent, allowing it to subsequently check and delete the  $+\text{WH}\{\text{-NOM}\}$  licenser on the second  $[\text{int}]$  head shown above.

## 4.5 The time complexity of the formalism

Harkema (page 107-108) shows that his CKY MG parser has an asymptotic time complexity of  $O(n^{4k+4})$ , where  $k = |\text{licensees}|$ . This complexity measure is for an MG without head movement, and was calculated as follows. Each MG expression has the form  $[(x_0, y_0) : \alpha_0, (x_1, y_1) : \alpha_1, \dots, (x_m, y_m) : \alpha_m]$ , where each  $(x_i, y_i)$  is some span and each  $\alpha$  is some feature sequence attached to that span. Each part (i.e. chain)  $(x_i, y_i) : \alpha_i$  has  $O(n^2)$  possible instantiations, because both  $x_i$  and  $y_i$  range between 0 and  $n$ ,  $n$  being the length of the sentence. The length of the string has no impact on the number of possible choices for  $\alpha_i$ , which is nevertheless bounded because the lexicon is finite, as are the feature sequences attached to lexical items; and since the feature sequences of any chain are suffixes of sequences appearing on lexical items, the number of choices

for  $\alpha_i$  is constant. Any expression can have at most  $k + 1$  parts:  $k$  moving chains (owing to SMC), and 1 head chain. This means there can be at most  $O(n^{2k+2})$  items in the chart at any one time. And because items which are merged during the course of the derivation need not appear adjacent to one another in the surface string, the parser must in principle try merging every item in the chart with every other item in the chart,<sup>40</sup> hence the time complexity of Harkema's CKY MG parser is  $O(n^{4k+4})$

The grammar and treebank presented in this dissertation were constructed using Autobank (see the next chapter) which uses a reimplementaion of Harkema's CKY MG parser that has been extended to incorporate all the additional rules and the selectional/agreement feature system presented in this chapter. Owing to the fine-grained selectional properties and requirements, which as we saw in section 3.4.2 can percolate up through the derivation tree, it is no longer true that the sequences of features on non-terminal MG types are necessarily suffixes of the sequences on lexical items. Nevertheless, the number of choices of these sequences is finite and constant (preserving the MCFG equivalence result), because each group of selectional properties and requirements attached to some structure building feature constitutes an unordered set, hence only one instance of each property/requirement (say +3SG or NOM) can be associated with any given structure building feature; for example, +CASE{+3SG.+3SG.NOM} is an impossible feature.

How large is  $k$ ? There are 17 different licensee features used in the MGbank grammar. These are: -case, D, -tough, -pers, -num, -epp, -self, -top, -foc, -wh,  $v\sim$ ,  $t\sim$ ,  $c\sim$ , -n, -loc, -negs, -pol. However, -pol and -negs are only permitted to trigger covert movement (see section 4.2.5), and the number of possible instantiations of any covertly moving chain does not depend on the length of the input because these chains have empty spans. As such, covert-only licensees are part of the constant and should not be included in  $k$ . We are then left with 15 licensees which are capable of triggering overt movement. However, as we saw in sections 4.2 and 4.2.6, DSMC places these licensees into 4 groups, and only one licensee from each group can be active in the derivation at any one time. As a result, the maximum number of moving chains with some overt string component in any given expression is 4. In addition, owing to the inclusion of head movement in the grammar, the head chain's string may consist of up to three non-empty substrings instead of one, which means that we must add 2 to our value of  $k$  in Harkema's equation, meaning that we have  $O(n^{4(k+2)+4})$  or, equivalently,

---

<sup>40</sup>In practice there are ways to avoid this, for example by organising each cell of the chart so that expressions whose head chains have the same first feature are grouped together

$O(n^{4k+12})$ . The worst case theoretical time complexity of the CKY MG parser used to construct MGbank is therefore  $O(n^{28})$ . The combination of the very strong grammar and the use of the PTB and CCGbank structures to prune the search space during parsing (discussed in the next chapter), nevertheless allowed this parser to operate efficiently enough to perform the treebanking task.

Fowlie and Koller (2017) show that in fact MG chart parsing is possible in  $O(n^{2k+3})$ , but with head movement this would increase to  $O(n^{2k+9})$  (Stanojević, 2019) which would mean that the MGbank grammar has a worst case time complexity of  $O(n^{17})$ . Stanojević (2019), however, demonstrates that MGs with head movement can be parsed in  $O(n^{2k+5})$  time. On this measure, the worst-case complexity of the MGbank grammar would be  $O(n^{13})$ .

Even the most optimistic of these complexity measures is clearly a very high order polynomial. It should be stressed, however, that these measures are very much the theoretical worst case, and in fact in chapter 6 we shall see that the wide-coverage parser presented there has a practical expected complexity of just  $O(n^3)$ , which it achieves via a good neural probabilistic model and A\* search, with the strong grammar also a likely contributor. Of course, the size of the grammar is also an important factor in the overall measurement of complexity, and MGs are known to be exponentially more succinct than the MCFGs with which they are weakly equivalent (Stabler, 2013b).

## 4.6 The expressive power of the formalism

As is discussed in the relevant sections, none of the extensions to MGs which are adopted in this paper extend the weak generative power of the formalism beyond that of MCFGs. In section 4.3.1.1 we saw that MGs with the Specifier Island Constraint have been shown to be equivalent to well-nested MCFGs, and hence cannot generate the MIX language and so are mildly context sensitive in the sense of Joshi (1985). As was discussed in that section, however, it did not prove feasible to enforce SpIC for constructions that involve an overtly expressed anaphor or associate. Therefore, the formalism has to allow for certain exceptions to SpIC which it does by allowing moving chains whose active feature is associated with an EDGE property to escape specifier islands. Therefore, the EDMG formalism presented here is equivalent to MCFGs, but not to well-nested MCFGs.

## 4.7 The fine-grained structure of clauses

We saw in section 3.2.8 that in Minimalism clauses are generally taken to have a tripartite structure consisting of the thematic domain (VP), the inflectional domain (TP) and the discourse domain (CP). From the early 1990s onwards it became standard practice to analyse each of these three domains as being decomposable into a more articulated structure consisting of multiple heads and their projections. We have already seen that VP decomposes into vP and VP layers. In the following three sections, we take a closer look at each of the three domains in turn.

### 4.7.1 The theta domain (VP)

In section 3.2.7 we saw that all arguments of verbs, including subjects, are in Minimalism (and MGBank) assumed to be base generated inside the verb phrase, where they are assigned their thematic roles (by checking =d/d= licensors). We also introduced the idea in section 3.2.8 that the VP decomposes into an inner VP core and an outer vP shell. This so-called VP-Shell Hypothesis was originally proposed in Larson (1988) (and first implemented for MGs in Hale 2003), where it was motivated by the need to integrate constructions featuring multiple complements into a strictly binary branching (Kayne, 1984) phrase structural framework. For instance, consider the following double object example.

(281) John gave [<sub>DP</sub> Bill] [<sub>DP</sub> a book]

The Penn Treebank treats such constructions as involving a ternary branching VP with the verb and the two objects as daughters. This is not an option in a strictly binary branching theory of phrase structure, such as the version of Xbar Theory used in GB or the Bare Phrase Structure theory of MP.

In the absence of a VP shell layer, one approach to 281 would be to suppose that the verb selects *Bill* as its complement argument, and *a book* as an inner rightward specifier (with the AGENT subject generated in the outer leftward specifier position). On these assumptions, this sentence would have the structure shown in fig 4.17, where to simplify the exposition we have omitted the (covert) movement of the objects for case-checking.

This analysis is in fact entertained in Chomsky (1981, page 171). There are good reasons, however, for thinking that it is incorrect. For instance, as we saw in section 4.2.4, reflexive anaphors such as *herself* must be bound by a (local) *c-commanding*

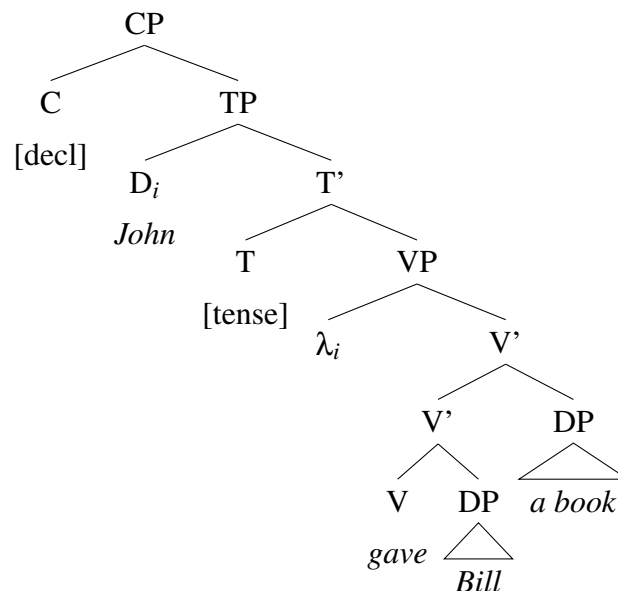


Figure 4.17: A possible analysis (entertained in Chomsky (1981, page 171)) for the sentence *John gave Bill a book* in which *a book* is analysed as a rightward specifier.

antecedent. Thus in the following example, *herself* must corefer with the larger DP *Sue's mother*, not with the small DP *Sue* embedded within it.

(282) [Sue<sub>i</sub>'s mother]<sub>j</sub> loves herself<sub>\*i/j</sub>.

In light of this, consider the fact that in the double object construction in 283, the leftmost object co-refers with the reflexive one.

(283) Jack showed Mary<sub>i</sub> herself<sub>i</sub> (in the mirror).

This suggests that the left object c-commands the right one, which in turn undermines the analysis in fig 4.17 because there the right argument is a specifier which c-commands the left one. Furthermore, it seems clear that a rightmost object cannot bind a leftmost reflexive one, as shown by the following.

(284) \*Jack showed herself<sub>i</sub> Mary<sub>i</sub> (in the mirror).

In fact, a standard assumption since at least the late GB period has been that specifiers (when defined so as to exclude functional heads like determiners and auxiliaries) are exclusively merged to the left of their heads. Word orders in which a head precedes its specifier must therefore be derived via movement. In support of this claim, Kayne (1994, page 35) observes that while there are categories of phrase for which spec-head surface ordering is cross-linguistically predominant, this is not the case for head-spec



ordering. CP is one example: spec-CP is the typical landing site for constituents which undergo wh-movement, and these items overwhelmingly appear in clause-initial position in languages with overt wh-movement. This was true even in earlier forms of English, when both the complementizer and the moved wh item could sometimes be overtly expressed (in violation of the Multiply Filled Comp Filter in operation over present day standard English - see section 4.7.6), as the following relative clause example from Chaucer (taken from (Radford, 2004, page 228)) indicates.

(285) In every peril [which that is to drede...]

(Chaucer, *Troilus and Criseyde*, circa mid-1380s)

Other arguments which Kayne adduces in support of his contention that rightward specifiers are impossible include the extreme cross-linguistic marginality of basic word orders in which objects precede subjects, and the lack of any so-called reverse V2 languages. In a V2 language such as German or Dutch, the finite verb must appear as the second top level constituent in a main clause. The clause initial position can be filled by either the subject, or by another topicalised constituent such as the object or an adjunct, in which case the subject appears post-verbally. For example, all of the following are possible translations of the English sentence *Jack must build the furniture quickly*.

- (286) a. Jack muss die M obel schnell bauen  
 Jack must the furniture quickly build
- b. die M obel muss Jack schnell bauen  
 the furniture must Jack quickly build
- c. schnell muss Jack die M obel bauen  
 Quickly must Jack the furniture build

The standard analysis in TG for V2 constructions is in terms of head movement of the finite verb to the C position with only the topicalised constituent in spec-CP preceding it. Kayne argues that if it were the case that specifiers could be rightward branching (and complements could be leftward branching), then it should be possible to have head-final languages with the mirror configuration in which the finite verb precedes only the topic in a sentence final rightward spec-CP position, contrary to fact.

Kayne was not just arguing against the possibility of rightward specifiers, but also against the existence of leftward complements, rightward adjuncts and rightward movement. All of these assumptions are required for his LCA theory of linearization, which MGBank does not adopt for reasons which were discussed in section 3.3.3.

Nevertheless, even those Minimalists who reject the LCA and allow for leftward complements, rightward adjuncts and rightward movements (notably Ernst (2002)), still usually assume that specifiers can only be leftward, perhaps for reasons having to do with information structure. The MGbank formalism therefore does not include any Merge or Move rules which would create rightward specifiers, meaning that the structure in 4.17 is simply not derivable in the present context.

Instead, MGbank adopts the standard Minimalist analysis of double object (and, more generally, double complement) constructions, according to which the VP domain decomposes into at least two layers: an inner VP core, and an outer vP shell. This is known as the VP-shell hypothesis and goes back to Larson (1988), although Chomsky (1993) refined Larson's original proposal by suggesting that the outer shell was headed by a null light verb referred to affectionately in the literature as *little v*. The main lexical verb begins in the V position, where it can select for a complement and one or more specifiers, these being the internal arguments of the verb, and subsequently undergoes head movement to adjoin to *little v* placing it to the left of both of its complements. In transitive (or unergative intransitive) structures, the *little v* head is a causativizing head which is responsible for selecting the external AGENT argument, which then (usually) moves to spec-TP. As we saw in section 3.2.9, in languages such as Kannada, this causative head is overtly expressed and shows up as a suffix on the verb.

Some authors assume that unaccusative verbs lack a *little vP* layer entirely (see, e.g., Hornstein et al. (2005, pages 102-103)). On the assumption that *little v* both generates the AGENT and is the head responsible for accusative case appearing on the object, this provides a neat explanation for Burzio's Generalization (Burzio, 1986), according to which verbs which are passive/unaccusative lack both an external argument and accusative case assigning capacity.

Chomsky (Chomsky, 2008, page 10) regards *little v* as the head responsible for determining the verbal category of the root it selects (i.e. the main verb), and in Chomsky (2001, 2008, 2012) distinguishes between transitive *little v* (which he refers to as *v\**) which selects a DP specifier and transmits its phi features (and hence its accusative case licensing capacity), and intransitive *little v*, which does not select for an AGENT and lacks case licensing features. A *little vP* layer is thus present in all clause types in Chomsky's system, transitive/unergative and unaccusative alike. The MGbank grammar adopts this perspective (although the case licenser is precompiled onto V in the lexicon, rather than being inherited from *v*) as doing so simplifies the grammar somewhat. For example, we do not need to duplicate all T heads to allow for

the two scenarios where T sometimes selects vP and sometimes selects VP directly. The following two null light verbs are therefore used in MGBank for transitive and unaccusative little v respectively.

[trans] :: >v= =d lv

[intrans] :: >v= lv

The analysis for our example sentence *John gave Bill a book* is given in fig 4.18. MGBank assumes that objects move overtly for case checking, and this is also represented in fig 4.18. Notice that the rightmost argument is now c-commanded by both of the other two arguments. This is consistent with the facts surrounding reflexive and reciprocal anaphor binding discussed above and in section 4.2.4.

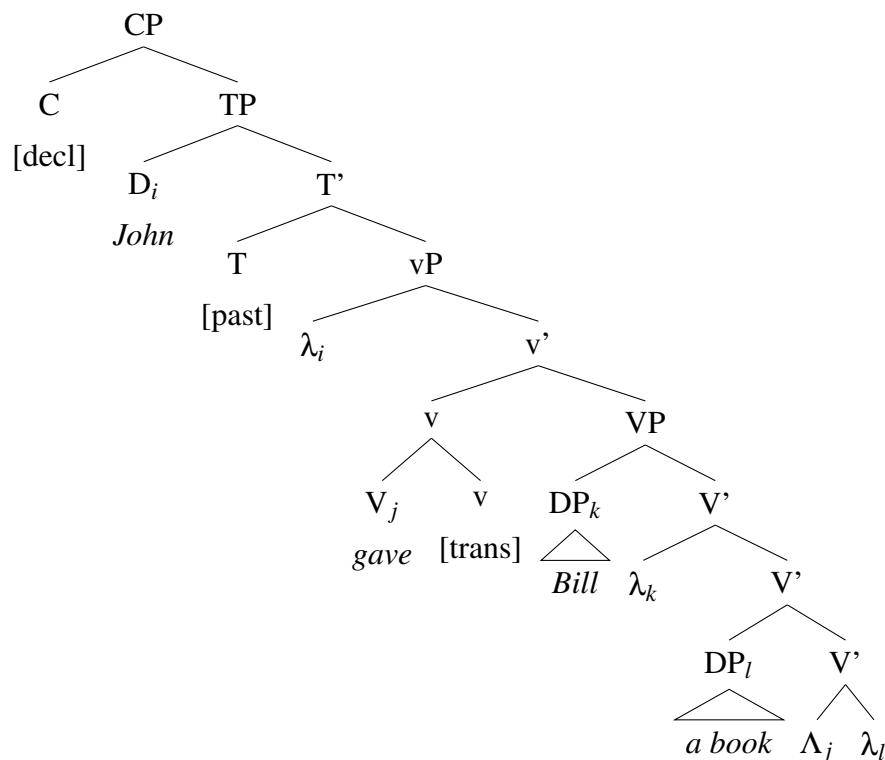


Figure 4.18: The shell structural analysis used by the MGBank grammar for double object sentence *John gave Bill a book*.

Notice that under this analysis the verb and its rightmost complement begin the derivation as a constituent which excludes the leftmost complement (and the subject), before subsequently becoming separated. We saw in section 3.2.7 that idiomaticity is often used as a diagnostic for deep constituency in TG. It is therefore interesting to note that there exist discontinuous idioms, such as *Jack took Mary to the cleaners*,

which consist of just the verb and the rightmost complement, and exclude the leftmost complement.

### 4.7.2 Case Theory and the Case Filter

Throughout this chapter we have seen instances of both subject and object DPs moving in order to check their *-case* licensees. In MGBank, all DPs, except NOC PRO, must check *-case*. These abstract movement operations are not found in the Penn Treebank or other more surface oriented formalisms. In this section, we will therefore look at some of the theoretical and empirical motivation for case-driven A-movement, with the aim of demystifying this aspect of the MGBank grammar.

In GB, the Case Filter (whose origins Chomsky credits to Vergnaud (1982)) was a Principle of the grammar responsible for constraining the distribution of DPs. Although case only shows up morphologically in the personal pronoun system in English, a standard assumption since GB has been that all (argumental) DPs in all languages have abstract case, regardless of whether or not this manifests itself in a language's overt morphology. The Case Filter required that all phonetically realised DPs be assigned (abstract) case by SS. If a DP was generated in a position where it could not receive case, such as the object position of a passive verb, it would have to raise to another position from which it could receive a case assignment or the sentence would be ungrammatical.

Instead of using separate construction-specific transformational rules for passive, unaccusative and raising structures, as in the earlier (Revised) (Extended) Standard Theory, GB's Case Theory unified these constructions via a single transformational rule, *move- $\alpha$*  (move anything anywhere), whose application was both driven and constrained by the Case Filter (along with various other Principles of the grammar, such as the Empty Category Principle). Consider the following examples.

- (287) a. She deceived him. (active)  
 b. He was deceived (by her). (passive)  
 c. \*It was deceived him. (impersonal passive)

Example 287a shows that the verb *deceive* in its canonical active form takes a THEME object argument and assigns accusative case to that object. Once passivised as in 287a, however, the verb by hypothesis loses its accusative case assigning ability, meaning that the object DP must move to the surface subject position to receive nominative case, though it retains its original semantic role as the logical object of the verb.

As 287c illustrates, this requirement cannot be circumvented by inserting the expletive subject *it* into the subject position (to satisfy the Extended Projection Principle, requiring all clauses to have subjects), despite the fact that this strategy is available for CP complements.<sup>41</sup>, as the following examples illustrate.

- (288) a. Jack believed that several people would help.  
 b. That several people would help was believed by Jack.  
 c. It was believed that several people would help.

Raising predicates, such as *seems* in 289 below, are superficially very different from passivised transitive verbs, and in the ST and its various extensions and revisions the two were accounted for using the separate transformational rules RAISING and PASSIVE.

- (289) He seems to have helped.

However, both types of verb share the property that they feature a DP subject which is not their logical subject. Unlike the passivised versions of transitive verbs, such as *deceived* in 287b, raising predicates do not select for a DP complement and then promote it to their subject position. Instead, they take an infinitival clausal complement and then raise a DP from inside that clausal complement to their surface subject position. That the subject in 289 is not a thematic subject argument of *seem* is supported by the fact that this surface subject position can instead be filled by an expletive, with the DP remaining in the lower clause, provided that the lower clause is finite, as in 290a. If the embedded clause remains infinitival, however, filling the surface subject position with an expletive results in ungrammaticality, as shown by 290b.

- (290) a. It seems that he has helped.  
 b. \*It seems him/he to have helped.

This situation contrasts with that of ECM predicates, such as *expect*, which are able to assign accusative case to the subject of their infinitival complements, which therefore need not raise to the matrix subject position, as shown by the acceptability of 291 below.

---

<sup>41</sup>CPs are often assumed to lack case, a generalisation which was formalised by Stowell (1981) as the *Case Resistance Principle*. Other authors have argued for a more nuanced version of this principle which allows CPs to sometimes bear case (see, e.g., Sheehan (2011)). In the MGBank grammar at present only DPs have -case licensees. However, there is a null [det] head which effectively maps a CP into a DP with a -case feature, and this is used to allow finite/infinitival CPs to be subjects of finite clauses, as in [*that you would do that*] *surprises me*, and [*to do that*] *would be folly*.

(291) She expects him to help.

GB explains these facts by postulating that, just as active verbs contrast with passive ones in terms of their case assigning ability, so too do finite and infinitival T, in that only the former is able to assign (nominative) case. This forces the DP subject of an infinitival clause to raise into a higher clause in order to receive a case assignment and satisfy the case filter, as in 289 (ECM constructions are analysed in MGBank as involving overt movement of the infinitival subject to spec-VP if the ECM verb to check accusative case following Chomsky (2008) - see section 4.7.3 below). However, if the surface subject position is already filled, the DP cannot move out of the infinitival clause and the result is an ungrammatical sentence, as shown in 290b.

The Case Filter thus provides a unified account of the distribution of the DPs in these examples, on the assumption that both passivised verbs and infinitival *to* lack the ability to assign case. This account was also generalised to other constructions, including unaccusative intransitive examples such as 292a below. Like passives and raising predicates, these types of unaccusatives are assumed in TG to lack case assigning/checking ability, with the result that their logical objects must move to become their surface subjects. That the DP occupies the object position at some point in the derivation is evidenced by example 292b, where the expletive pronoun *there* occupies the surface subject position, forcing the DP to remain in situ (see section 3.3.5 on how the DP has its case feature checked in existentials such as 292b).

- (292) a. Several problems will remain.  
 b. There will remain several problems.

In the transition to Minimalism, the internal levels of representation DS and SS were eliminated and many of the stipulative grammar-internal filters which had been stated over them were recast as arguably more natural *bare output conditions*, which require that any LF and PF objects be interpretable at the C-I and A-P interfaces. Chomsky (1995) assumes that DPs enter the derivation already fully specified for case features such as nominative or accusative, but must have these features checked and deleted (rather than assigned) by moving to the specifier of an appropriate head. Structural case is assumed to be uninterpretable at LF, and so any DP which fails to have its case feature checked and deleted before LF will cause the derivation to crash and result in a grammatically deviant sentence. The basic Case Theoretic approach to constraining the distribution of DPs remains essentially intact from GB, however, and this

early MP case checking approach<sup>42</sup> is the one adopted by MGBank.

In Minimalism, as we have seen, finite T is the head responsible for checking nominative case, which causes the subject to raise to spec-TP overtly in English. There is much more disagreement over the exact nature of accusative case checking of the object in transitive constructions, however, with some authors arguing that this case is checked by V and others arguing that it is checked by v. Furthermore, because case now need only be checked by LF rather than by SS (the latter having been eliminated in the transition from GB to MP), this opens up the possibility that accusative case can be checked via covert movement/Agree. A further point of disagreement in MP therefore concerns whether the object undergoes overt or covert A-movement to spec-VP or spec-vP to check A-features.

Chomsky (2008) suggests that V may inherit uninterpretable phi-features from v (just as he assumes that T inherits its uninterpretable phi features from C) before triggering overt movement of the object to spec-VP. Because V then undergoes V-to-V head movement, it correctly surfaces to the left of the object. The MGBank grammar follows Chomsky in that it assumes accusative case checking to be carried out by V rather than v, and that such checking triggers overt rather than covert movement of the object to spec-VP (The inheritance of A-licensors from v to V is not modelled, however, as this would require non-trivial and non-monotonic extensions to the formalism, because the object would need to move to spec VP *after* v has taken this VP as its complement (thereby violating the Extension Condition)).

The (simplified) structure of a simple transitive sentence such as *they read it* is shown in fig 4.19, where the subject moves from spec-vP to spec-TP to check nominative case, while the object moves from comp-VP to spec-VP to check accusative case.

### 4.7.3 Infinitival clauses and successive cyclic A-movement

One advantage of assuming that accusative case checking involves *overt* movement to spec-VP is that this allows for a simple and elegant account of so-called Exceptional

---

<sup>42</sup>In the later Agree-based approach, Chomsky assumes case to be the feature which activates a DP as a goal which can be probed by a head that is in need of having its phi features valued (with associated EPP features driving any accompanying A-movement), with different case morphologies simply being a reflex of phi-feature agreement with different heads (e.g. T vs V). As is standard practice in the MG literature, the formalism presented here adopts the earlier perspective in that it uses +/-case licensor and licensee features to drive A-movement, although it does bundle agreement and specific case properties (nominative vs accusative etc) onto those features, as we saw in section 3.4. See also section 292b on the MGBank approach to so-called *Multiple Agree*.

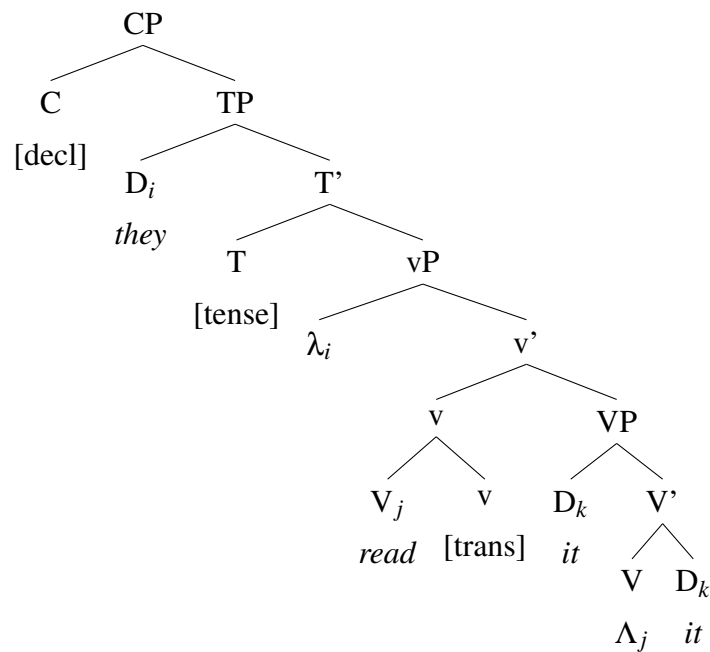


Figure 4.19: An analysis for the sentence *they read it* with overt A-movement of both the subject and object for case checking purposes.

Case Marking (ECM) constructions, which were historically quite challenging for both GB and Minimalism. Infinitival clauses in English generally seem to require a null subject. As we saw in the previous section, in the case of the infinitival complements of subject raising verbs such as *seem*, these null subjects are in MP standardly analysed as traces of case/phi-driven A-movement. Control clauses such as *Jack wants to have helped* also feature a null subject in the infinitival clause which is coreferential with the subject of the higher clause, and as we saw in section 3.3.6 this null subject is analysed as a null pronominal by some Minimalists and as a trace of theta-driven A-movement by others; as we saw in that earlier section, MGBank adopts the latter perspective. What is important for current purposes, is that in general infinitival T and finite T differ in that only the latter generally appears to license an overt subject in its specifier, and we saw above that this led GB researchers to propose that *to* lacks the case assigning abilities of its finite counterpart.

However, consider the following ECM example.

(293) He expected her to help.

Here *her* appears immediately left-adjacent to *to* and is clearly not a thematic argument of the matrix verb, as the thing expected is not *her* but the entire proposition expressed by the complement clause; rather, *her* is clearly just an AGENT argument



of the lower verb. That this is the correct analysis of the theta assignment properties of ECM verbs is evidenced by the fact that they can take expletive *there* as their object.

(294) He expected there to be help.

This contrasts with superficially similar object control verbs, such as *persuade*, which assign both case *and* a theta role to their object, and hence cannot select for expletive *there*.

(295) \*He persuaded there to be help.

The traditional assumption in GB and earlier Minimalism at least was that ECM objects occupy the specifier position of the TP headed by infinitival *to* from where they are assigned case *exceptionally* across a clause boundary. In order to avoid the problem that CP generally constitutes a (phase) barrier to operations such as case assignment/checking, the clausal complements of ECM (and raising) predicates are standardly analysed as bare TPs lacking the CP layer. This classical analysis of ECM constructions is shown in figure 4.20.

The problem with this analysis is that it is not clear why a DP should be able to move to spec-*to* and stop there when this is apparently not possible in other infinitival constructions, as illustrated for the raising verb *seem* in 296 below.

(296) \*There seems a man to be outside.

It may seem tempting to suppose that ECM *to* is a special variant of the infinitival particle, which unlike its raising counterpart is able to check the accusative case feature of its specifier. However, on closer inspection it becomes clear that in fact it is the ECM verb which checks this case: once we passivise *expect* and thereby eliminate its case checking ability, the ECM object is forced to move to the matrix subject position to check nominative case, as illustrated below.

(297) a. \*It was expected her to help.  
b. She was expected to help.

In effect then, passivizing an ECM verb like *expect* transforms it into a raising verb like *seem* (i.e. it knocks out its ability to assign an external theta role and to license accusative case). If ECM objects do not move to the spec-*to* for case checking, the challenge is then to explain why they move there at all. Furthermore, whereas all other instances of case assignment appear to take place in highly local head-complement or

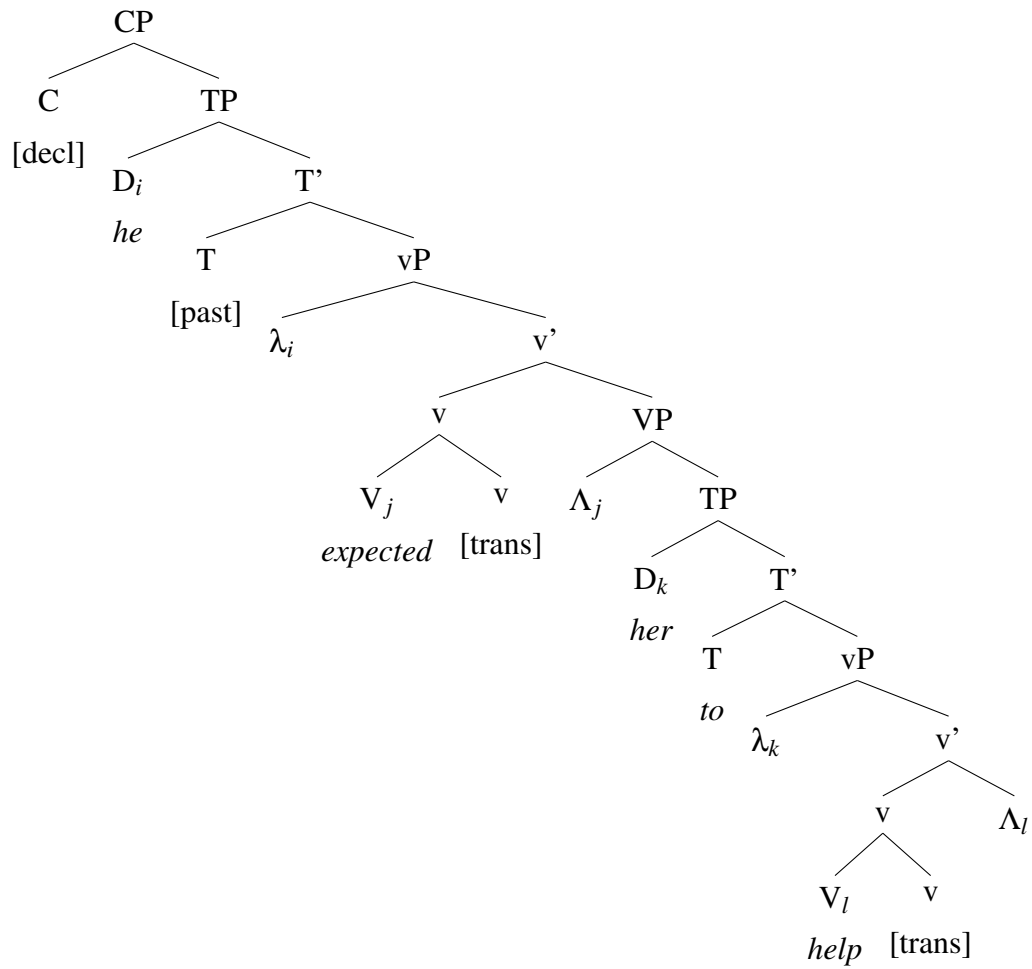


Figure 4.20: A classical analysis of ECM constructions in which the infinitival subject is a specifier of the infinitival TP.

spec-head configurations (which in Minimalism are assumed to ‘come for free’ given the conceptual necessity of Merge), the assumption that ECM objects are located in the downstairs clause would appear to call for some longer distance relation in the grammar under which case could be checked/assigned. In GB, this was a major motivating factor behind the generalisation of the head-comp relation to the long-distance *government* relation. This in turn led to various ad hoc theoretical devices such as the *barriers* and *blocking categories* of Chomsky (1986a), which were designed to prevent government across the CP clause boundaries embedded under control verbs and verbs like *say*, while allowing it into the TP complement clauses of ECM verbs and raising predicates.

In Minimalism both government and barriers have been eliminated, but in their place we find covert case assignment (or, in later Minimalism, the long-distance operation Agree) and the notion of the *phase*. As we saw in section 4.3.4, phases are chunks of structure (specifically, CP and transitive vP in Chomsky (2001b)) which are transferred to the PF and LF interfaces cyclically. Any constituents inside the CP can neither escape that CP nor have their features checked after that CP phase is transferred to the interfaces, unless they are situated in the phase *edge*, i.e. in spec-CP at the time of the transfer. This then allows for the same escape mechanism for the complement clauses of both ECM and raising predicates that was used in GB, as these are still generally considered to lack the CP layer and hence not to constitute phase barriers. This allows the ECM object to be assigned case in its downstairs spec-TP position by the ECM verb (and for a DP to raise out of the complement clause of a raising predicate).

However, we still have not accounted for the fact that a DP appears to move to spec-TP of ECM infinitival complement clauses and stop there, but cannot do so in the infinitival complement clause of a raising predicate. We have seen that this movement is unlikely to be to check case. Chomsky (1995, 2001b, 2004) argued that this movement was the result of some other morphosyntactic feature checking requirement of *to*. The licensor feature in question is sometimes referred to generically as the EPP feature, which is a nod to the Extended Projection Principle of GB which required all clauses to have a subject. Like most principles, the EPP essentially just restated the empirical facts, rather than explaining them.

As we have seen, Minimalism attempts to better explain these sorts of phenomena by reducing them to the arguably more natural requirement that all syntactic objects must be interpretable at the interfaces. Since the EPP feature is by hypothesis uninterpretable (though whether at PF or LF would depend on its precise nature), it must

be deleted at some point during the syntactic derivation. If infinitival *to* carries an EPP feature, therefore, the movement of a DP to *spec-to* could be explained by the need to delete this feature. The assumption is then that this movement occurs in the infinitival complements of both ECM verbs and raising predicates, but that in the case of raising predicates the DP must continue raising to the surface subject position of a finite clause to check nominative case (as raising predicates, unlike ECM predicates, are unaccusative and hence cannot check accusative case).

However, as noted in (Epstein and Seely, 2006, page 50), the precise identity of this EPP feature has never been entirely clear. In Chomsky (1995) it is claimed to be a D-feature, while in Chomsky (2001b) it is an uninterpretable person feature, and later still in Chomsky (2004) it is described as an ‘edge’ feature. Epstein and Seely (2006) argue at length that in fact the EPP feature is problematic in various respects and should therefore simply be eliminated. Part of their discussion centres around our earlier example 296 repeated as 298 below.

(298) \*There seems a man to be outside.

Epstein and Seely (2006) argue that Chomsky’s (2001b) attempt to block examples such as this engenders a ‘domino effect’ of problems. For example, Chomsky (2000) proposes that example 298 be barred owing to an economy constraint which he calls Merge-over-Move (MoM). This states that whenever both Merge and Move could in principle apply, Merge must be selected over Move. This is because Move is regarded as a complex constraint consisting of the two sub-operations Copy and Merge. Thus at the following derivational point, either Move or Merge could in principle apply: *a man* could move to *spec of to* to check the latter’s EPP feature, or *there* could be Merged directly into this position as a pure EPP checker.

(299) \_ [to be [a man outside]]

Since Merge trumps Move wherever both may apply, only the latter operation is permitted, yielding the following derivation stage.

(300) there [to be [a man outside]]

Subsequently, *there* will raise to its final surface subject position to check the EPP feature of the matrix finite T head. However, one problem with the Merge-over-Move analysis is that it is then necessary to explain why *a man* is able to move all the way to the subject position in 301 below.

(301) A man seems to be outside.

The derivation of the above example begins precisely as before up to the following derivational step.

(302) \_ [to be [a man outside]]

However, in this case *a man* by hypothesis first moves to *spec-to* and then moves again to its final surface position in *spec* of the matrix finite TP. The problem is to explain why *there* is not once again inserted to check the EPP feature of *to*, given that Merge is by hypothesis preferred over Move. As Epstein and Seely (2006) note, this issue in part motivates the numeration: if *there* is present in the numeration at the start of the derivation for 298 but not 301, then it can only be inserted into *spec-to* in the former case. However, as Chomsky notes, this numeration-based analysis cannot by itself account for example 303 below.

(303) There is a possibility that proofs will be discovered.

In this case *there* is clearly present in the numeration, and yet at the following derivational step the system must choose to move *proofs* to *spec* of the finite TP rather than merging *there* into this position.

(304) \_ [will be discovered proofs.]

Chomsky's solution to this is to propose that each phase is associated with its own separate numeration, or subarray. The embedded clause CP phase thus has its own separate subarray which excludes *there*, meaning that this item is not available for Merge at the derivational step in 304. In the case of example 298, however, the embedded clause is the complement of a raising predicate, hence it lacks a CP layer and is thus not a phase. The same is true of the unaccusative vP of the matrix clause, since Chomsky assumes only transitive vP (which he calls v\*P) to constitute a phase. This means that there is just a single subarray for the entire sentence in this case and *there* is thus available to prevent *a man* from raising to *spec-to*.

As we saw in section 4.3.4, Chomsky's conception of phases (along with their purported correspondence to cyclic points of access to the lexicon) has been criticised by various authors. Epstein and Seely (2006) argue that subarrays/phases and the problematic Merge-over-Move constraint can all be dispensed with simply by assuming that raising/ECM *to* does not attract a specifier at all. This immediately blocks the problematic example 298 above: *a man* cannot move to *spec-to* because *to* has no features to check. What about the problematic ECM example 293, repeated as 305 below?

(305) He expected her to help.

Epstein and Seely (2006) propose, following in part Johnson (1991a), Koizumi (1995), and Lasnik and Saito (1991), that in fact the notion that the ECM object appears in *spec-to* is actually an ‘optical illusion’ and that this DP has actually undergone object shift. In the same year Chomsky (2008) adopted such a ‘raising to object’ analysis of ECMs in which the ECM object moves to *spec-VP* of the upstairs clause. However, Epstein and Seely’s proposal differs from both Chomsky’s and those of the other authors cited in that they argue that the ECM object not only does not stop in *spec-to* but that it does not even pass through it. MGBank adopts an analysis whereby the ECM object moves directly to *spec-VP* of the ECM verb, and this is illustrated in fig 4.24.

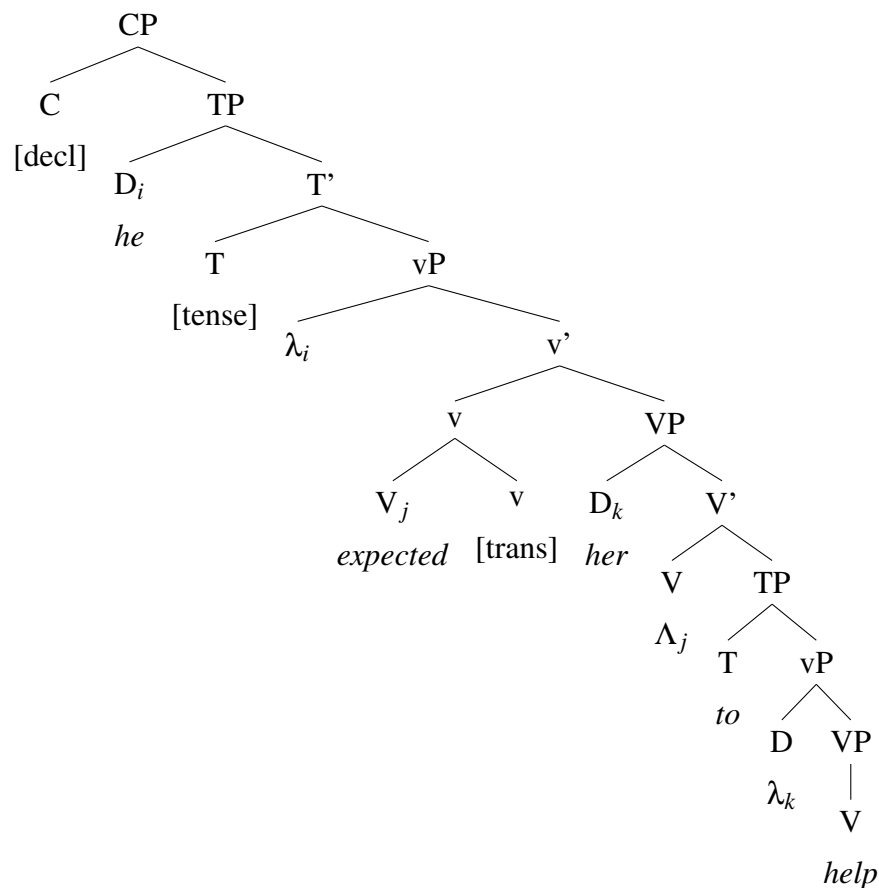


Figure 4.21: Derived Xbar tree for the sentence *he expected her to help* showing the raising to *spec-VP* analysis for ECMs used in MGBank.

The assumption that verbs check case via overt movement of the object to *spec-VP* thus allows for a relatively simple analysis of the word order facts in ECM construc-

tions. If object shift were covert, we would again be in the position of having to assume that the ECM object first moves to *spec-to* (or at least *somewhere* to the left of *to* but below the VP), before covertly moving to check case in *spec-VP*. Furthermore, assuming that object shift was overtly to *spec-vP* would place the object to the left of the verb, which would necessitate an additional head movement of the V-v complex head to some higher head. There is ample evidence from adverb placement that main verbs in English do not move to T as they do in French for example, so we would need to introduce another null head below T but above *v* to serve as the target for this additional head movement in all transitive clauses. One of the design methodologies adopted for the construction of MGBank was to use as few null heads and ad hoc movement operations as possible, and so this approach was eschewed in favour of the simpler one. Of course, even if other transitive verbs checked case covertly, there would be nothing preventing us from allowing ECM verbs from being exceptional in the sense of being the only verbs which check case overtly. However, it is clearly preferable to avoid such ad hoc exceptions wherever possible.

As noted above, Epstein and Seely (2006) argue that A-movement does not move through *spec-to*, at least for raising and ECM constructions. In such constructions, A-movement is thus hypothesised to proceed in one fell swoop rather than successive cyclically. This is true even where A-movement proceeds across multiple clauses, as in 306 below.

(306) We<sub>*i*</sub> are likely [<sub>*TP*</sub> to be asked [<sub>*TP*</sub> to t<sub>*i*</sub> build airplanes]].

MGBank adopts Epstein and Seely's proposal that A-movement does not proceed successive cyclically via *spec* of raising/ECM *to*, and also extends this to control *to* which it regards as one and the same item. In fact, it turns out that this swoop analysis of case/phi-driven A-movement has a further advantage in view of MGBank's adoption of the Movement Theory of Control (MTC) (see section 3.3.6). To see this, consider the following example.

(307) We<sub>*i*</sub> wanted to seem to try to [<sub>*vP*</sub> t<sub>*i*</sub> help]

This sentence features two control verbs, *want* and *try* straddling the raising verb *seem*. We have already seen that the MTC requires that for control, A-movement must proceed successive cyclically through intermediate *spec-vPs* to pick up each successive theta role. This was achieved by allowing D to optionally persist after being checked. But consider what would happen if the DP in question also had to move successive cyclically via *spec-to* in order to check some -*epp* licensee.

(308) [<sub>TP</sub> We<sub>i</sub> [<sub>vP</sub> t<sub>i</sub> wanted [<sub>TP</sub> t<sub>i</sub> to [<sub>vP</sub> seem [<sub>TP</sub> t<sub>i</sub> to [<sub>vP</sub> t<sub>i</sub> try [<sub>TP</sub> t<sub>i</sub> to [<sub>vP</sub> t<sub>i</sub> help]]]]]]]]].

In the analysis indicated here, movements for theta checking of D/=d are interspersed with non-thematic movements for EPP checking. Because MG is more restrictive than MP in the sense that it requires features to be strictly ordered, in order to generate this analysis we would require the following type with multiple D and -*epp* features.

D -*epp* D -*epp* D -*case*

And of course, we can go on stringing control and raising clauses together without bound which would mean we would require an infinite number of types with interspersed D and -*epp* features to adequately handle all possible cases. One solution would be to assume that *to* also checks D, rather than -*epp* (in Chomsky (1995) the EPP feature is referred to as a D feature). However, we would then lose the nice correspondence between theta role assignment and =d/D checking, as spec-TP is generally considered to be a non-thematic subject position. Instead, MGBank simply assumes with Epstein and Seely (2006) that *to* checks no features and hence does not attract a DP specifier.

One problem for the swoop analysis of case/phi-driven A-movement proposed in Epstein and Seely (2006) is posed by the existence of *for-to* constructions such as 309 below.

(309) Jack asked [for Mary to help].

Here the subject of the bracketed infinitival clause appears to the left of *to* but to the right of the complementizer *for* so that the raising-to-object strategy that was adopted for ECMs cannot be pursued here.

In such examples, *for* is sometimes referred to as a *prepositional* complementizer for two reasons: firstly, it has the same phonological shape as the pure preposition *for* from which it derives diachronically (see Lightfoot (1976) for the relevant historical data); and secondly, it appears to be responsible for assigning accusative case to the infinitival subject, prepositions being canonical case assigners. The MGBank grammar interprets the claim that *for* is a prepositional complementizer by adopting a shell structure for this element whose inner layer is a PP and whose outer layer is a CP. Complementizer *for* starts out in the lower P position where it assigns case to the subject, triggering overt movement of the latter to its spec. The PP which it projects is then



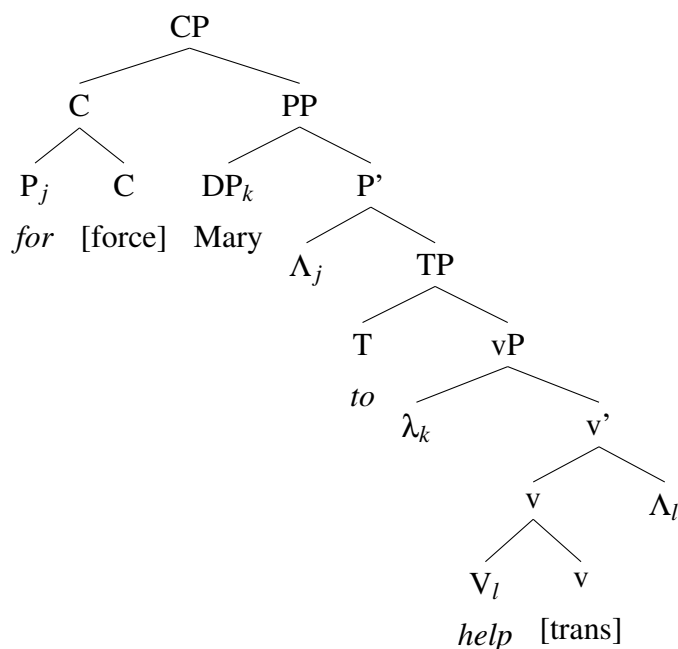


Figure 4.22: MGBank's CP-PP shell analysis for the *for-to* clause *for Mary to help*.

selected for by a null C head which triggers P-to-C head movement of *for*, correctly placing the latter to the left of the infinitival subject. This allows us to maintain Epstein and Seely's claim that infinitival *to* does not attract a specifier. The derived Xbar tree for this analysis of the bracketed clause in 309 is given in fig 4.22.

#### 4.7.4 Non-obligatory Control

We saw in section 3.3.6 that MGBank adopts a movement based approach to obligatory control (OC), and in the last section we saw that this decision lead us to also adopt the proposal by Epstein and Seely (2006) that infinitival *to* checks no features. These decisions had further knock-on effects in MGBank with respect to its treatment of non-obligatory control (NOC), the details of which are discussed in this section.

NOC PRO in many ways exhibits precisely the opposite characteristics to OC PRO, as the following pairs of examples, taken from (Boeckx et al., 2010, page 196), illustrate. 310a shows that OC PRO requires an antecedent, while 310b shows that NOC PRO does not; and even if NOC PRO does have an antecedent, that antecedent need neither be local nor in a c-command configuration with NOC PRO (311b and 312b respectively) again in contrast with OC PRO (311a and 312b respectively).

(310) a. \*It was expected PRO to shave himself.

- b. It is illegal PRO to park here.
- (311) a. \*Jack<sub>i</sub> thinks that it was expected PRO<sub>i</sub> to shave himself.  
 b. Jack<sub>i</sub> thinks that Mary said that PRO<sub>i</sub> shaving himself is vital.
- (312) a. \*Jack<sub>i</sub>'s campaign expects PRO<sub>i</sub> to shave himself.  
 b. Jack<sub>i</sub>'s friends believe that PRO<sub>i</sub> keeping himself under control is vital if he is to succeed.

In fact, OC and NOC PRO appear to be in more or less complementary distribution with one another. For these reasons, Hornstein (2001) and Boeckx et al. (2010) analyse NOC PRO as a null pronominal, rather than as a trace of A-movement. More specifically, they argue that NOC PRO is a null resumptive pronoun which only gets inserted as a last resort if movement is not available owing to a grammatical economy constraint according to which movement is more economical than pronominalisation. This explains why NOC PRO typically shows up in island environments. For example, in both 311b and 312b, PRO is nested within both a subject-island and a nominative clause island and yet the coreference is still possible.

One aspect of NOC PRO which Boeckx et al. (2010) do not discuss is whether it has a case feature to check. Hornstein (2001) argues that NOC PRO is the 'little pro' which appears in pro-drop languages such as Spanish in spec-TP of finite clauses, which is canonically a nominative case licensing position. However, Boeckx et al. (2010) argue at length against the idea, first proposed in Chomsky and Lasnik (1993) and subsequently more fully worked out in Martin (1996, 2001), that OC PRO has a null case feature which is checked by control *to*. Furthermore, Boeckx et al. (2010, pages 201-202) also argue that structures themselves should not be classified as NOC or OC, but only the *relations* which hold between the nominal expressions contained within them; in other words, there are no OC or NOC clauses, only OC or NOC relations between nominals. This would seem to imply that they regard NOC *to* and OC *to* as one and the same item, meaning that *to* in NOC constructions cannot check case. This is in line with MGBank's treatment of infinitival *to* as being the same lexical item across all construction types.

If *to* cannot check case in NOC control structures, this implies that NOC PRO does not have a -case licensee to check. Otherwise, the following example should be ungrammatical, contrary to fact.

- (313) To wash oneself is advisable.

In GB, NOC/OC PRO was considered to be a pronominal anaphor (the so-called *pro theorem* of Chomsky (1981)) and therefore subject to both Principle A (an anaphor must be bound within its binding domain) and Principle B (a pronominal must be free in its binding domain). This created a paradox since PRO was required to be both free and governed within its binding domain. Chomsky concluded that the only way to satisfy both constraints was for PRO to not have any binding domain at all. As we saw in section 4.2.4, the binding domain of a given constituent  $\alpha$  was defined in Chomsky (1981) as being the smallest clause containing  $\alpha$ , a governor for  $\alpha$ , and a SUBJECT, where SUBJECT includes the traditional notion of a subject. Chomsky therefore argued that infinitival TP (IP) could not be a governor (in contrast to finite TP) and that PRO, which was assumed to be located in infinitival spec-TP, was therefore ungoverned and hence did not have a binding domain. One consequence of this fact was that NOC/OC PRO lacked case, as case could only be assigned under government. This was argued to explain its lack of phonetic content, since overt DPs were considered to require case marking by SS.

The MGBank approach to NOC PRO inherits the GB perspective on NOC PRO a null caseless pronoun. Of course, OC PRO, regarded here as an A-movement trace, is also caseless in MGBank, because all A-movement traces are necessarily located in caseless positions. From the strong derivational perspective adopted here, however, traces are not substantive syntactic objects, but merely convenient notational look back devices - see section 3.2.5.

We can incorporate NOC PRO into our grammar using the following caseless null D category.

[pro-d] :: D

Although Boeckx et al. (2010) assume that control *to* cannot check case, it seems clear that they assume it does check *some* feature(s) of NOC PRO, because as illustrated in examples 310b, 311b and 312b, they represent NOC PRO as appearing to the left of *to*, implying that it has moved to spec-*to* from its base-generated position in spec-vP. We saw in section 4.7.3 that allowing any version of *to* to check A-features is problematic for examples involving interleaved control and raising verbs. MGBank therefore assumes that NOC PRO does not move to spec-*to* to check any features, but simply stays in its base-generated position in spec-vP, unless of course it has to undergo movement to establish an OC relation as in example 314 below.

(314) To [<sub>vP</sub> PRO<sub>i</sub> try to [<sub>vP</sub> t<sub>i</sub> wash oneself is advisable]].

As noted, Hornstein (2001) and Boeckx et al. (2010) assume that the complementary distribution of OC and NOC PRO arises from a derivational economy constraint whereby movement is more economical than pronominalisation. Call this constraint, Move over Pronominalisation (MoP). Because pronominalisation is not an operation of the grammar presented here (it is simply precompiled into the lexicon), it was not possible to implement this constraint directly. Nevertheless, this constraint is currently approximated during parsing using two strategies. The first strategy involves holding certain null heads back from the chart unless the parser cannot get a parse without them. One important consequence of this is that the number of types of movement which the parser can try is initially limited because topicalisation, focalisation and rightward movement, for instance, are all triggered when a null [topicalizer], [focalizer] or [extraposer] head applies to, say, a DP in order to transform it from a regular DP into one which undergoes the relevant movement type.

The more types of movement we allow the parser to entertain, the greater its search space becomes and hence the slower the parsing will be. During the generation of MGbank it was found that allowing all null heads into the chart at once resulted in impractically slow parse times for certain sentences. Therefore, a strategy which was adopted was to hold certain null heads back initially and only introduce them incrementally if the parser did not find a full parse without them. The null heads are introduced by the parser according to the name of their empty string. So, at a certain point, all [focalizer] heads are added to the chart simultaneously, after which all [topicalizer] heads are added, and so on. The null head types which are held back from the chart initially are given in table 4.1 in the order in which they are introduced.

Of course, one drawback to this strategy is that an incorrect analysis may end up bleeding the correct one if the correct one includes a null head which is only introduced at a later stage from the null heads contained in the incorrect one. It was therefore important to carefully engineer the grammar to be very constrained so that it blocked as many of these incorrect analyses as possible. Note that this drawback does not apply when parsing with the complex supertag categories to be introduced in chapter 6, since in this case all null heads are anchored to an overt head inside complex LTAG-like categories and the parser is free to introduce any of these complex categories whenever it likes. On the other hand under this approach MoP is no longer enforced via the strategy of holding certain null heads back from the chart. In any event, this first strategy for enforcing MoP is only approximate and will not work for sentences

null head name	description
caseless [pro-d]	NOC PRO.
[relativizer]	causes an NP to undergo relativization movement.
all other [pro-x] categories	e.g. [pro-v] heads used for VP ellipsis and cased 2nd person [pro-d] subject of imperatives.
[sbj]/[-sbj]/[epp]/[float]	[sbj], [-sbj] and [epp] are used in locative inversion, [float] in floating quantifier constructions.
[extraposer]	the heads triggering rightward movement.
[focalizer]	the heads triggering focus movement.
[topicalizer]	the heads triggering topic movement.
[op]	used in tough movement constructions.
[appositizer]	used for certain adjunctions instead of [adjunctizer].

Table 4.1: A list of null heads which are initially held back from the chart to make parsing more efficient. The items are presented in the order in which they are introduced into the chart.

which require any of the null heads introduced after NOC PRO in fig 4.1. Consider the following example.

(315) That guy<sub>j</sub>, we<sub>i</sub> [<sub>vP</sub> t<sub>i</sub> wanted to [<sub>vP</sub> seem to [<sub>vP</sub> t<sub>i</sub> try to [<sub>vP</sub> t<sub>i</sub> help t<sub>j</sub>]]]]

On our current assumptions, this example involves successive cyclic control movement of *we*, first from spec-vP of the *help* clause to spec-vP of the *try* clause, and then on to the matrix spec-vP position (bypassing spec vP of the unaccusative raising verb *seem*) and finally to the matrix spec-TP position. However, because this example also involves topicalisation, and because [topicalizer] heads are introduced *after* caseless [pro-d] which is therefore already present in the chart, the parser could potentially propose a whole host of bogus analyses, shown below, in which spec-vP of the *help*, *try* and/or *want* clauses is filled by NOC PRO (note that at least one spec-vP must be a trace in order to base generate *we*).

- (316) a. That guy<sub>j</sub>, we<sub>i</sub> [<sub>vP</sub> t<sub>i</sub> wanted to [<sub>vP</sub> seem to [<sub>vP</sub> t<sub>i</sub> try to [<sub>vP</sub> PRO help t<sub>j</sub>]]]]  
 b. That guy<sub>j</sub>, we<sub>i</sub> [<sub>vP</sub> t<sub>i</sub> wanted to [<sub>vP</sub> seem to [<sub>vP</sub> PRO try to [<sub>vP</sub> t<sub>i</sub> help t<sub>j</sub>]]]]  
 c. That guy<sub>j</sub>, we<sub>i</sub> [<sub>vP</sub> PRO wanted to [<sub>vP</sub> seem to [<sub>vP</sub> t<sub>i</sub> try to [<sub>vP</sub> t<sub>i</sub> help t<sub>j</sub>]]]]  
 d. That guy<sub>j</sub>, we<sub>i</sub> [<sub>vP</sub> t<sub>i</sub> wanted to [<sub>vP</sub> seem to [<sub>vP</sub> PRO try to [<sub>vP</sub> PRO help t<sub>j</sub>]]]]  
 e. That guy<sub>j</sub>, we<sub>i</sub> [<sub>vP</sub> PRO wanted to [<sub>vP</sub> seem to [<sub>vP</sub> PRO try to [<sub>vP</sub> t<sub>i</sub> help t<sub>j</sub>]]]]  
 f. That guy<sub>j</sub>, we<sub>i</sub> [<sub>vP</sub> PRO wanted to [<sub>vP</sub> seem to [<sub>vP</sub> t<sub>i</sub> try to [<sub>vP</sub> PRO help t<sub>j</sub>]]]]

The reason why these analyses are possible is that NOC PRO lacks case in our grammar. This in turn means that it can be merged into an intermediate spec-vP position without triggering a (D)SMC violation, allowing *we* to simply skip over it. Since

NOC PRO need not be co-indexed with *we*, this will incorrectly lead to some semantic ambiguity and will also unnecessarily increase the search space of the parser, leading to decreased efficiency.

We saw in section 4.7.3 that we do not need to adopt Chomsky's Merge-over-Move constraint because we are following Epstein and Seely (2006) in assuming that there is no A-movement to spec-*to*. We also saw in section 4.2.4 that, somewhat ironically, certain structures involving the binding of reflexives necessitate the adoption of a locally determined Move-over-Merge Constraint. This constraint turns out to be very useful for NOC PRO, enabling the system to implement a stronger version of MoP and avoid many of the bogus analyses in 316, specifically 316b, 316c, 316e and 316f. Consider the following stage in the derivation of 315 above.

(317)  $\_ [{}_{vP} \text{ try to } [{}_{vP} \text{ we help}]]$

Here, the parser could in principle either Move *we* to spec-*vP* to check *try*'s =d feature or it could Merge caseless [pro-d] into that position to achieve the same thing; but because Move is available, it will only perform this operation, blocking the Merge option.

Unfortunately, Move-over-Merge does not help with examples 316a and 316e, because in these instances PRO is introduced into the derivation before *we*, meaning that the latter is not available to undergo Move and therefore bleed Merger of PRO. For the purposes of the treebank generation task, most remaining bogus NOC PRO analyses were weeded out using the 'disprefer [pro-x]' metric discussed in section 5.4.4.2 of chapter 5, which deducts a point from a candidate parse for every null proform head which it includes. As far as realistic wide-coverage parsing is concerned, we could also look to the statistical model to guide the parser towards the correct distribution of OC vs NOC.

#### 4.7.5 The inflectional domain (TP)

Just as the VP domain has been decomposed into more than one distinct projection, so too, following the influential work on verb movement of Pollock (1989), has the inflectional TP (formerly IP) domain. The most extensive work on this area is cartographic approach Cinque (1999, 2006). Cinque presented evidence for a cross-linguistically invariant ordering of functional heads within the TP domain, arguing, for example, that in languages with tense morphemes and epistemic mood morphemes, the former

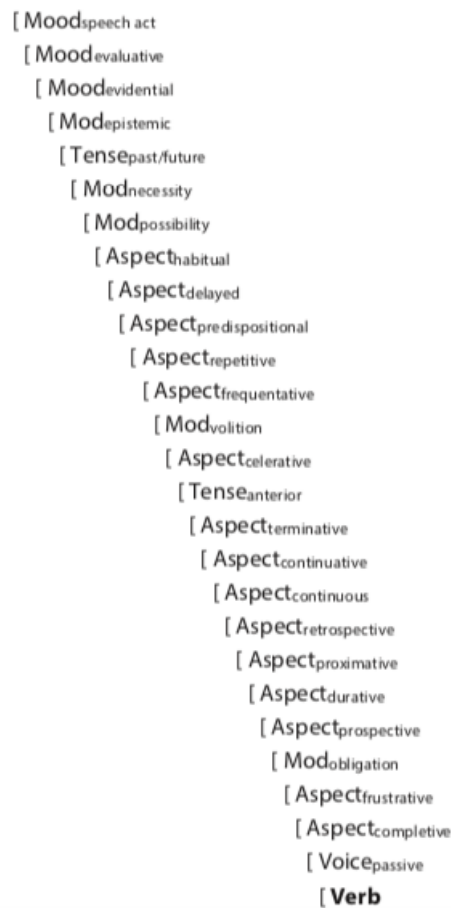


Figure 4.23: Part of Cinque's (1999) universal hierarchy of functional heads.

invariably occur closer to the verb; on the other hand, continuous aspectual morphemes occur even closer to the verb than tense morphemes across languages.

Cinque also argues that the universal hierarchy of clausal heads is mirrored by the permissible ordering of different types of adverbs across languages. Allowing for the sometimes obscuring effects of various types of movement (e.g. the displacement of VPs), Cinque proposes an extremely articulated clausal hierarchy of some 30 functional heads, each of which can in principle host an adverb of the relevant type in its specifier. A language which expresses epistemic mood may therefore do so either with an overt epistemic morpheme or with an epistemic adverb as the specifier of a null epistemic head (Cinque is assuming a Kaynian architecture in which adjuncts and specifiers are treated identically from a structural point of view). Part of Cinque's hierarchy is shown in fig. 4.23.

Cinque's hierarchy is intended to be generalisable to all human languages. The goal of the MGBank project is the somewhat more modest one of providing a descriptively

accurate description of English. As such, it was not necessary to include anywhere near 30 distinct heads in the grammar, and a much reduced set of 7 is used instead; this set consists of two negation heads, a tense head, one modal head, two aspect heads and a voice head. Cinque also assumes that all of the heads in his hierarchy are always present in any given construction in any given language, regardless of whether they are overtly realised in the language or construction in question. Even with the much reduced set of heads used in MGBank, including all of them in every clause would have made for extremely large and difficult-to-read trees. For this reason, the only four projections which are included in all (non-defective) clauses are CP, TP, vP and VP, with TP thus being the only obligatory head within the inflectional domain. Other heads are included only as and when needed, as is standard practice in the MP literature.

One admitted drawback of this approach is that it increases the size of the lexicon because it is necessary to include multiple versions of each head for every possible complement type it can take. For example, there are separate versions of T for the scenarios where it takes vP, voiceP, progP, perfP, modP or negP as its complement.

The 7 heads used in the TP domain in MGBank are given below in the fixed order in which they are permitted to occur.

Neg(ation)    T(ense)    Mod(al)    Neg(ation)    Perf(ect)    Prog(ressive)    Voice

The reader may be curious as to why there are two separate negation heads included. The MGBank analysis of negation is taken from (Radford, 2004, pages 170-183) (the idea that negation heads its own projection goes back to Pollock (1989)). The positioning of the lower negP below modP but above perfP is intended to account for the fact that example 318a below means roughly ‘it is the case that he must not go’ whereas 318b means roughly ‘it is not the case that he has gone’; in other words, the modal auxiliary *must* scopes over negation in 318a, whereas the perfective auxiliary *have* is scoped over by negation in 318b.

- (318) a. He must not go. (the modal auxiliary scopes over the negation)  
       b. He has not gone. (the negation scopes over the perfect auxiliary)

This difference is in spite of the fact that both auxiliaries precede the negation, and is accounted for by the fact that in their base positions, the perf is c-commanded by the lower neg head, whereas mod c-commands the lower neg head.

Note that the neg head itself is actually null in MGBank, and takes the overt negative particle *not* or *n't* in its specifier. The reader is referred to Radford (2004, pages 170-



173) for arguments in support of this analysis based on Middle English examples such as 319 below, in which there were two overt negative particles (just as there are in present day French, for example), one of which (*ne*) by hypothesis fills the overt head position of negP while the other (*nat*) fills the spec-negP position.<sup>43</sup>

(319) A lord in his household ne hath nat every vessel al of gold. (Chaucer's *Wife of Bath's tale*, lines 99-100)

We saw above that in general, modal verbs scope over negation in contrast to other auxiliaries which are scoped over by negation. However, as noted by Cormack and Smith (2000) (cited by Radford (2004, page 173)), this is not the case for negative interrogatives such as 320 below, which is paraphrasable as 'is it not the case that you should be at work?' with negation scoping over the modal.

(320) Shouldn't you be at work? (negation scopes over the modal auxiliary)

Radford suggests that this fact can be accounted for if we assume that there is a second negP above TP and that in such sentences *not* is in fact the specifier of this higher negP projection. This analysis is supported by the possibility of sentences such as 321 below, which feature two instances of *not*, each of which can be hosted in the specifier of one of the two negPs.

(321) Mightn't he not have seen her?

Fig 4.24 shows the derived Xbar tree for the sentence *mightn't he not have been being hindered?* This example is included in the MGBank corpus because it features all 7 of the heads of the inflectional domain. It is standardly assumed in Minimalism that the finite auxiliary (here *should*) undergoes head movement to the T position, and because this is an interrogative, the T-mod complex also undergoes further *roll-up* head movement to C picking up the intervening null neg head along the way. Notice that the perfect auxiliary *have* also undergoes head movement to the lower neg head position. This occurs because this neg head triggers head movement of the head of its complement blindly in MGBank, in case that head is a finite auxiliary or copula *be*, in which case it would need to pass through this position on its way to T. In this particular instance, *have* is in its non-finite bare form (because it heads the complement of a modal verb), and so there is no further head movement beyond this position here.

<sup>43</sup>The finite verb undergoes roll up head movement to T via neg, to which it right adjoins, carrying the latter with it and placing it to the left of *nat*.

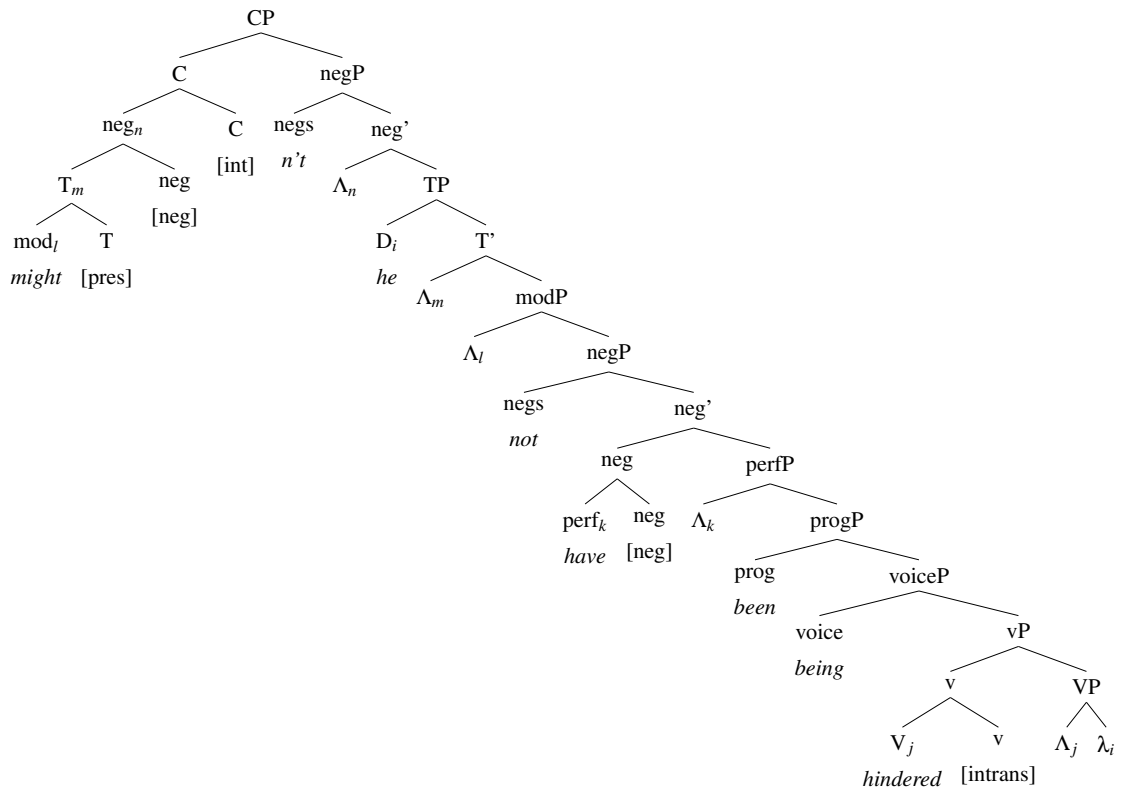


Figure 4.24: A derived Xbar tree for the sentence *mightn't he not have been being hindered?* which features all 7 major heads of the TP domain used in MGBank.

It is also worth noting that the strict ordering of different types of auxiliary is fully encoded in the MGBank lexicon. This is possible because, as the tree in fig 4.24 implies, each auxiliary has its own unique selectee feature which is only ever selected for by heads which occur higher up in the hierarchy. For example, the lexicon contains the following versions of the progressive and perfective auxiliaries.

*be* :: lv{+PROG}= prog{BARE}  
*be* :: voice{+PROG}= prog{BARE}  
*is* :: lv{+PROG}= prog{PRES.3SG}  
*is* :: voice{+PROG}= prog{PRES.3SG}  
*am* :: lv{+PROG}= prog{PRES.1SG}  
*am* :: voice{+PROG}= prog{PRES.1SG}  
*are* :: lv{+PROG}= prog{PRES.2SG.1PL.2PL.3PL}  
*are* :: voice{+PROG}= prog{PRES.2SG.1PL.2PL.3PL}  
*was* :: lv{+PROG}= prog{PAST.1SG.3SG}  
*was* :: voice{+PROG}= prog{PAST.1SG.3SG}  
*were* :: lv{+PROG}= prog{2SG.1PL.2PL.3PL}  
*were* :: voice{+PROG}= prog{2SG.1PL.2PL.3PL}  
*been* :: lv{+PROG}= prog{PERF}  
*been* :: voice{+PROG}= prog{PERF}  
*have* :: lv{+PERF}= perf{BARE}  
*have* :: voice{+PERF}= perf{BARE}  
*have* :: prog{+PERF}= perf{BARE}  
*has* :: lv{+PERF}= perf{3SG.PRES}  
*has* :: voice{+PERF}= perf{3SG.PRES}  
*has* :: prog{+PERF}= perf{3SG.PRES}  
*have* :: lv{+PERF}= perf{PRES.1SG.2SG.1PL.2PL.3PL}  
*have* :: voice{+PERF}= perf{PRES.1SG.2SG.1PL.2PL.3PL}  
*have* :: prog{+PERF}= perf{PRES.1SG.2SG.1PL.2PL.3PL}  
*had* :: lv{+PERF}= perf{PAST}  
*had* :: voice{+PERF}= perf{PAST}  
*had* :: prog{+PERF}= perf{PAST}

Notice that while for each phonological form of the perfective auxiliary there is a version which selects for progP, there are conversely no versions of the progressive

auxiliary which select for the perfect auxiliary. This ensures that perfective *have* always c-commands progressive *be*, and therefore avoids unwanted overgeneration of, e.g., *\*he is have helping/helped*. Furthermore, the fine-grained selectional requirements attached to each entry's selector feature ensure that the head of the complement of each of these heads will have the correct phonological form. For example, only the *been* form of progressive *be* is able to be selected for by perfective *have*, owing to the +PERF selectional property associated with the latter's prog= selector in the relevant entries; this avoids the overgeneration of sentences such as *\*he has being hindered*.

Finally, observe that while there are only two versions of each phonological form of the progressive auxiliary (one which selects for vP and one which selects for voiceP), there are three versions of each phonological form of the perfective auxiliary (selecting for vP, voiceP or progP). Indeed, the number of versions of each phonological form for a given head increases by one as we move up through the clausal hierarchy. This is the multiplication effect in the lexicon that was noted above as being a consequence of the decision taken here not to take the Cinquean approach of including all possible heads in all clauses regardless of whether they are overtly expressed.

One way of avoiding this duplication, without adopting Cinque's assumption, would be to give all auxiliary verbs the same selectee feature, say *t*, and then use fine-grained (disjunctive) selectional requirements and properties of the type described in section 3.4 in order to restrict the allowable orderings.

#### 4.7.6 The discourse domain (CP)

It has been persuasively argued that, at least for certain clause types, the CP domain, like the VP and TP domains, decomposes into several distinct projections. For example, Rizzi (1997) suggests that the complementizer system provides an interface between the internal propositional content of a clause (expressed by TP and its contents) and the superordinate structure, which is either a higher clause or, in the case of root clauses, the discourse context. For example, the complementizer *that* restricts its TP complement to being finite, whereas the complementizer *for* requires its TP complement to be infinitival. At the same time, *that* specifies that the clause it heads has declarative force, whereas the complementizer *whether* specifies interrogative force. Rizzi argues on the basis of data primarily from Italian that the force and finiteness properties expressed by the complementizer system are sometimes encoded on two separate Force and Finiteness heads.

As well as determining the outward and inward looking selectional properties of a clause, the CP domain also plays host to various preposed constituents, including topicalized and focalized constituents. Rizzi argues that we should also therefore distinguish separate topic and focus heads in the left periphery, which host topicalized and focalized constituents respectively in their specifiers, and that these heads should be sandwiched between the force and finiteness heads.

To illustrate the difference between topicalization and focalization, consider the following two examples taken from Rizzi (1997).

(322) [Your book]<sub>i</sub>, you should give t<sub>i</sub> to Paul (not to Bill)

(323) [YOUR BOOK]<sub>i</sub>, you should give t<sub>i</sub> to Paul (not mine)

In example 322, *your book* is the topic and the material following it is the comment. The topic typically represents old information which is already salient in the discourse, while the comment represents new information. By contrast, in the formally similar example 323, *your book* receives focal stress and represents new information, with the rest of the sentence expressing a presupposition, i.e. knowledge which the speaker presupposes to be shared with their interlocutor. In Italian, the topic-comment construction involves a resumptive clitic pronoun coreferential with the topic, whereas no such pronoun is required for the focus-presupposition construction (see Rizzi (1997, page 286) for the relevant data from Italian). In English, on the other hand, the only property differentiating these two different types of A'-movement in the above examples is focal stress, making it difficult if not impossible for a parser to capture,<sup>44</sup> and indeed, this distinction is not drawn in the Penn Treebank. The sentence *your book you should give to Paul* would therefore only receive the topicalization analysis by the MGBank grammar.

On the other hand, there are other types of focalization in English which are easily identifiable because they involve inversion of either an auxiliary or a main verb, as these are treated as involving focus features rather than topic features in MGBank. This type of focus movement includes quotative inversion (324), negative inversion (325), and inversion structures involving *so* (326).

(324) [Never before] have I seen such a mess.

(325) ["Compare two candidates for Mayor,"] says the announcer.

<sup>44</sup>Most automatic speech recognition systems do not currently output information about focal stress, so capturing this distinction is not generally feasible even when parsing from speech rather than text.

(326) [So often] are government statistics revised, that they seem to resemble a spinning weather vane.

Rizzi proposes the following articulated structure of the CP domain, also known as the *left periphery* of a clause.

Force      (Topic\*)      (Focus)      (Topic\*)      Finiteness

As indicated by the parentheses, the Topic and Focus heads are proposed to only be present when needed (i.e. when hosting a topicalized or focalized constituent in the relevant head's specifier), whereas Force and Finiteness are argued to always be present (as every clause has force and finiteness properties). Note also that there are two separate Topic positions either side of the Focus head and that these are associated with a Kleene star indicating that they can iterate without bound.<sup>45</sup> As far as wh-movement is concerned, Rizzi argues, again on the basis of Italian data, that relative wh-operators appear in spec ForceP, while relative question wh operators (being quantificational) appear in spec FocP.

Evidence that separate Topic and Focus heads should also be distinguished in English comes from the fact that it is possible to have separate topicalized and focalized constituents in the same clause, as example 327 below illustrates.

(327) He said that [that kind of behaviour], [never again] would he tolerate.

In this example, *that kind of behaviour* is clearly old information which is salient in the discourse, hence a topic, whereas *never again* is new, focused information. Notice too that the force head *that* precedes both the topic and focused constituents. Given our assumption that specifiers always occur to the left of their heads, neither the topic nor the focus can be in spec-forceP, meaning that we require three separate heads here. Combining Rizzi's proposals with our assumptions about the structure of TP, the embedded sentence would have a structure along the lines of that shown in fig 4.25.

As noted in section 4.2.2, however, DSMC prevents the generation of examples requiring two A'-movers in the derivation at the same time. Examples such as 327 are therefore currently undergenerated by the MGbank grammar. It is, however, difficult to find examples in real world corpora which require any sort of articulated left periphery, meaning that very little coverage is lost in this respect.<sup>46</sup>

<sup>45</sup>See Rizzi (1997, 295-296) for examples from Italian with multiple topicalized constituents appearing either side of the single allowable focalized constituent. These would not be allowed here owing to (D)SMC of course.

<sup>46</sup>From a Chomskyan perspective, this has nothing to do with the correctness of the grammar of course.

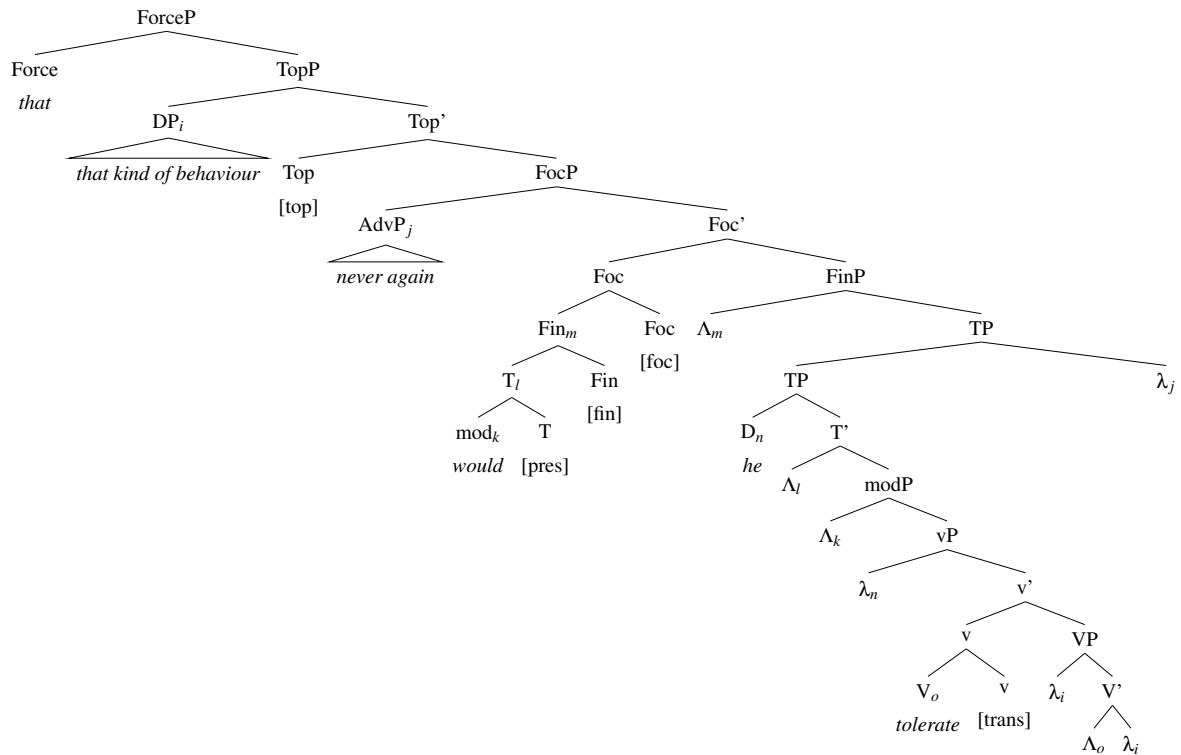


Figure 4.25: Rizzian-style structure for the embedded clause *that that kind of behaviour never again would he tolerate*, which contains topicalized and focalized constituents after the declarative complementizer.

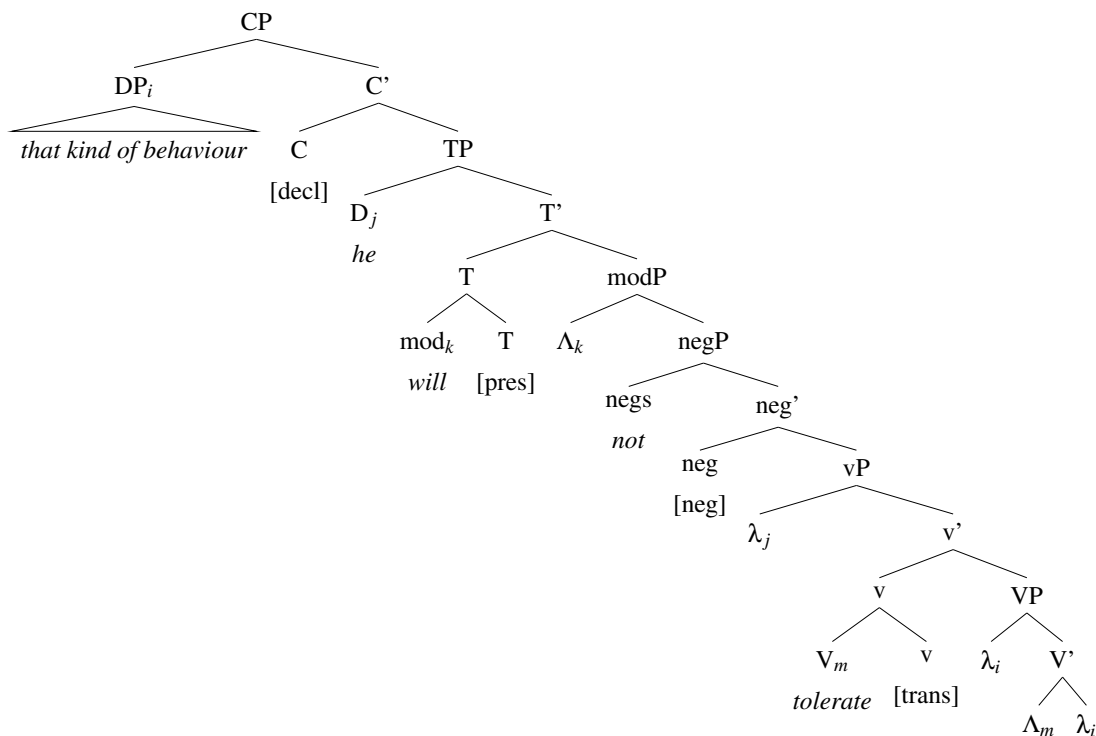


Figure 4.26: Derived Xbar structure for the main clause *that kind of behaviour he will not tolerate* with topicalization of *that kind of behaviour* to spec-CP.

Rizzi argues that in the absence of any topicalized or focalized element in the left periphery, the topic and focus heads are absent altogether and the Force and Finiteness heads are syncretised into a single CP head. The MGBank grammar goes a step further than this: even when a topic or focused element is present, the MGBank grammar usually uses a single C head with any +FOC, +TOP or +WH licensors which are required simply packed onto it (separate C heads are therefore required for each type of A'-movement). So, for instance, a simple topicalization sentence such as 328 below would have the structure shown in fig 4.26.

(328) That kind of behaviour, he will not tolerate.

There are a few cases in English where more than one head is required in the left periphery, although all such examples in MGBank were added manually as I could find none in the Penn Treebank. As we saw in example 327 above, it is possible for an embedded clause with an overtly expressed complementizer to feature both topicalized and focalized phrases in the left periphery. Even though MGBank cannot capture cases where *both* of these operations take place in the same clause (owing to DSMC), it can



capture sentences such as 329 and 330 below, where either one occurs by itself.

(329) He said that [that kind of behaviour] he would not tolerate.

(330) He said that [never again] would he tolerate that kind of behaviour.

In either case, the declarative force head *that* occurs to the left of the bracketed topicalized or focalized constituent meaning that the latter cannot be situated in the former's specifier position (because we are assuming with the majority of (even non-Kaynian) Minimalists that specifiers occur exclusively to the left of their heads - see section 4.7.1). In order to accommodate these constructions, MGBank includes separate null topic and focus heads, along with separate versions of the complementizer *that* (as well as of its null counterpart) which select for these heads. The relevant categories are given below (note the head movement diacritic on the Focus head which triggers the subject-auxiliary inversion).

*that* :: top= c  
*that* :: foc= c  
 [top] :: t= +TOP top  
 [foc] :: >t= +FOC foc

Notice that instead of using a force selectee for the complementizer as per Rizzi's analysis, we have retained the c selectee for the highest head in the left periphery. The reason for doing so is that otherwise we would need to include additional versions of every item in the lexicon which is able to select for a declarative CP complement, creating versions which instead select for forceP. For example, we would need an additional version of each phonological form of the verb *say* which selected for forceP rather than CP. The structures assigned to 329 and 330 in MGBank are given in figs 4.27 and 4.28 respectively.

There is also some evidence from English to support Rizzi's proposal that separate Finiteness and Force heads must sometimes be distinguished in the left periphery. Consider the following two examples, taken from (Radford, 2004, page 334).

(331) speaker A: What was the advice given by the police to the general public?

speaker B: (i) That [under no circumstances] should anyone approach the escaped convicts.

(ii) [Under no circumstances] for anyone to approach the escaped convicts.

Both of these examples feature a focalized constituent (in parentheses) adjacent to a complementizer. However, whereas the focalized constituent appears to the right of

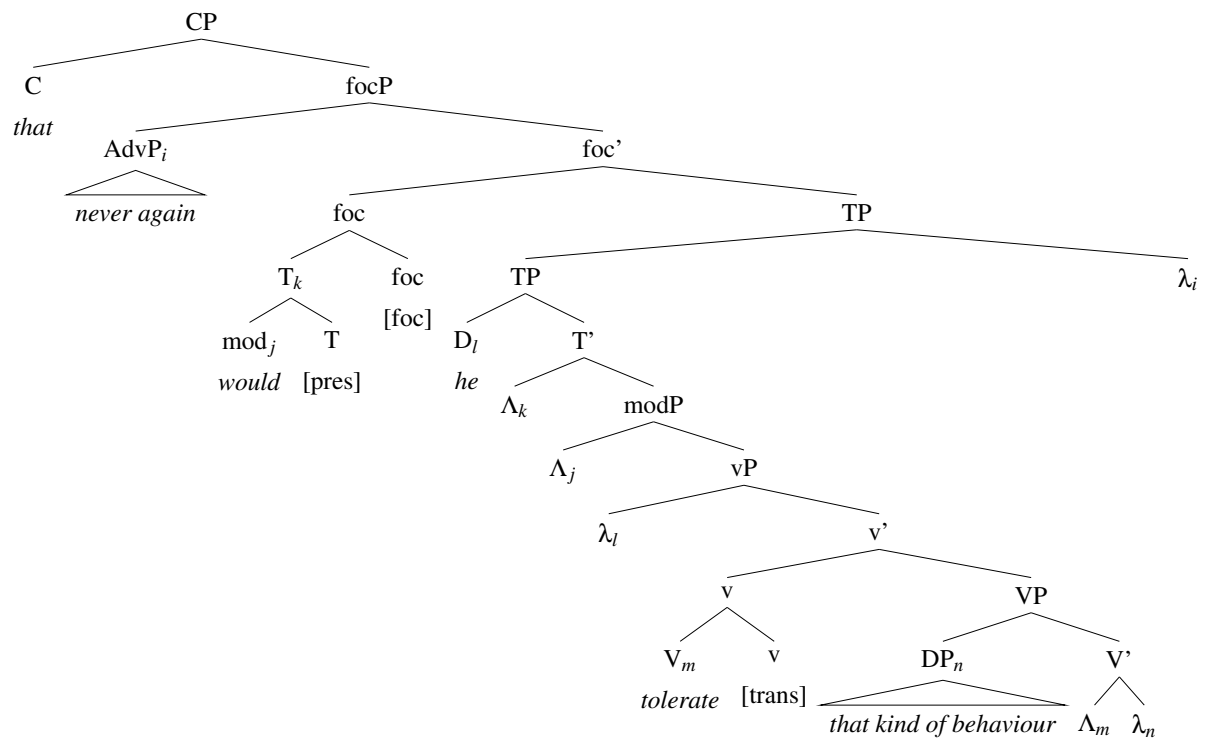


Figure 4.27: Derived Xbar tree for the embedded clause *that never again would he tolerate that kind of behaviour* which requires separate force (here C) and focus heads.

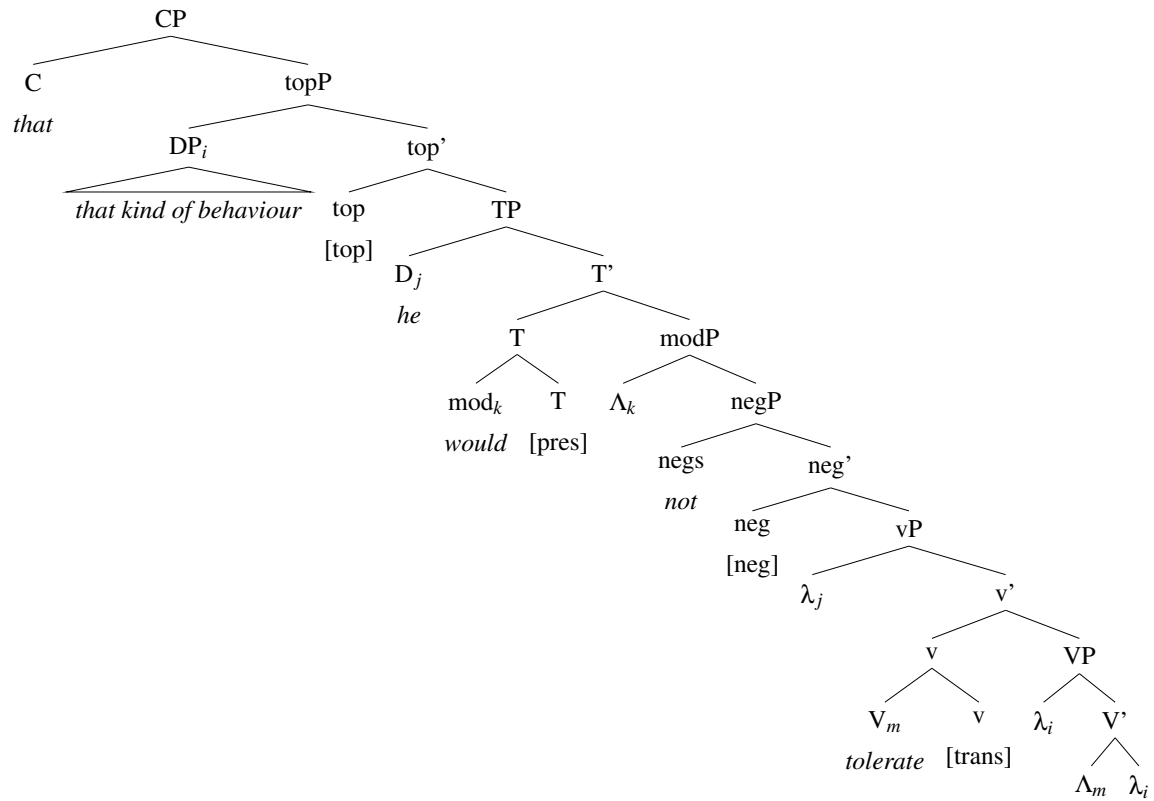


Figure 4.28: Derived Xbar structure for the embedded clause *that that kind of behaviour he would not tolerate* which requires separate force (here C) and topic heads.

the finite complementizer *that* in answer (i), it appears to the left of the non-finite complementizer *for* in answer (ii). Assuming that the focalized constituent is situated in spec-FocP in both examples, this implies that these two complementizers are the heads of different phrases. The obvious choice in the context of Rizzi's proposals is for *that* to head forceP and *for* to head finP. To accommodate examples like this, the MGBank grammar includes a null C head which attracts a topic and selects a finP complement headed by *for* (this is achieved using +FOR and FOR fine grained selectional features). Notice that because the focalized constituent appears to the left of the complementizer in for-to constructions we could simply assume that the left periphery is syncretized into a single C head that takes the focalized constituent as its specifier. One reason for not doing this is that there appears to be a general constraint in operation in English which Chomsky and Lasnik (1977) refer to as the Multiply Filled COMP Filter, and which (Radford, 2004, page 230) states informally as follows:

(332) Multiply Filled COMP Filter.

Any CP which contains an overt complementizer (*that/if/for*) with an overt specifier is ungrammatical.

This filter is intended to account for contrasts such as the following.

- (333) a. Get me something [<sub>CP</sub> that I can write with].  
 b. Get me something [<sub>CP</sub> with which I can write].  
 c. Get me something [<sub>CP</sub> which I can write with].  
 d. \*Get me something [<sub>CP</sub> which that I can write with].  
 e. \*Get me something [<sub>CP</sub> with which that I can write].

Rizzi (1997) provides very persuasive evidence from Italian that relative wh operators occupy spec-forceP (in contrast to wh question operators which occupy the lower spec-focP position). We are assuming here that in English *that* is the same item in both declarative and relative clauses (see section 4.2.1), and examples 327 and 331 suggest that this complementizer is the head of forceP, the uppermost projection in Rizzi's system. As the examples in 333 illustrate, *that* cannot take an overt wh-operator as its specifier, an effect described by the constraint in 332.

Interestingly, 332 also appears to hold for the infinitival complementizer *for*, as illustrated by the examples in 334 below, even though example 331 suggests that *for* heads finP not forceP.

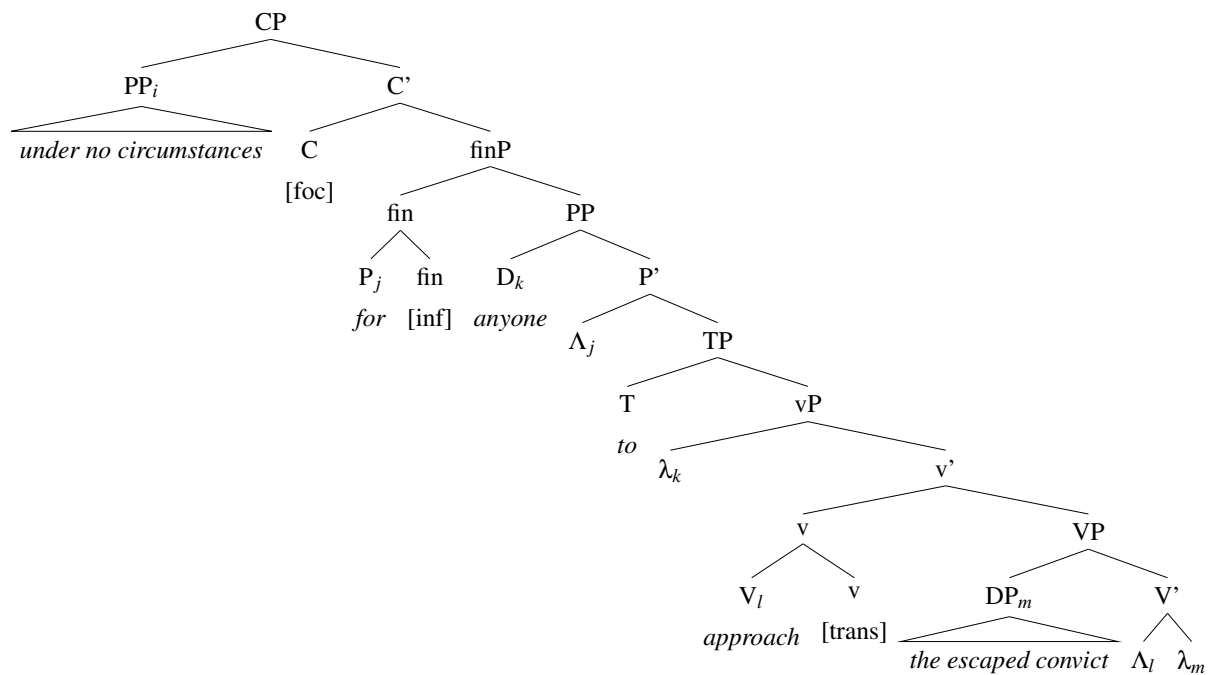


Figure 4.29: The MGBank analysis of the embedded infinitival clause *under no circumstances for anyone to approach the escaped convict*.

- (334) a. Get me something [<sub>CP</sub> for me to write with].  
 b. Get me something [<sub>CP</sub> with which to write].  
 c. \*Get me something [<sub>CP</sub> with which for me to write].

Speaker B's answer (ii) in 331 therefore seems to be a special case, as here *for* is seen with a focused item apparently in its specifier. To account for this, we will assume here that whereas in the examples in 334, the C domain is syncretised into a single head, meaning that there is only one specifier position in the left periphery, in Speaker B's answer in 331, the left periphery is (for reasons which are unclear) able to remain partially split, so that the focused element is not actually located in the specifier of the complementizer *for* at all, but is instead located in some higher head (perhaps focP, or the syncretism of focP and forceP, which we will simply label CP here).

The analysis of the infinitival clause in speaker B's answer (ii) in example 331 is shown in fig 4.29 (see section 4.7.3 for details of the MGBank CP-PP shell analysis of *for-to* infinitival clauses that is used in this structure).

# **Part II**

## **Performance**



## **Chapter 5**

### **Autobank and MGbank: semi-automatically constructing a wide-coverage MG treebank**



## 5.1 Introduction

Natural language is notoriously ambiguous, and for realistically sized grammars, even short sentences can potentially have a great many possible syntactic structures. Most applications require a single parse to be returned for each sentence, meaning that a parser must be equipped with the means to choose between alternative analyses. Furthermore, local indeterminacies during parsing can lead to an explosion in the search space, producing many search paths which ultimately lead nowhere but which can cause parsing to be very inefficient. The MGbank grammar presented in chapters 3 and 5 is strong, in the sense that it was designed not only to provide detailed analyses for as wide a range of (both frequently and rarely occurring) constructions as possible, but also to block as many illicit derivations as early on as possible. This reduces the amount of global and local ambiguity in the system and helps to keep parsing more efficient than would otherwise be the case. Nevertheless, for many sentences, a strong grammar by itself is insufficient. This is because, as we saw in chapter 1, there are many cases where the human grammar licenses an incorrect analysis which is nevertheless disregarded on the basis of world knowledge.

As also noted in chapter 1, the seminal work of Collins (1999) and Charniak (2000) demonstrated that statistical models extracted from gold standard treebanks can serve as proxies for such world knowledge and enable machines to accomplish the disambiguation task reasonably well. A statistical model component has in fact by now become a virtual necessity for any efficient wide-coverage parsing system. The estimation of these models can be done in a supervised or unsupervised fashion, where supervised approaches use labelled data in the form of a corpus of parse trees (known as a treebank), while unsupervised approaches use unlabelled data, such as a corpus of raw sentences. Supervised parsing systems currently outperform unsupervised ones significantly, and so the creation of treebanks for different formalisms and languages is an important task within NLP. An important contribution of the current project was the creation of the first wide-coverage MG treebank, called MGbank. MGbank remains a work in progress, but it has already been successfully used to train the first ever wide-coverage statistical MG parser, which is presented in the following chapter. The present chapter provides details on the treebank and the method used to construct it.

## 5.2 Semi-manual construction of treebanks

One approach to constructing new treebanks, which is optimal if time and money are not a consideration, is to do so largely manually. The Wall Street Journal section of the Penn Treebank (PTB) (Marcus et al., 1993), for instance, the most commonly used treebank within NLP, was created by a team of linguists, who inspected the output of a parser for each of nearly 50,000 sentences, selected the best candidate parse from this output, and finally made any structural changes necessary to perfect the parse. This process took around three years at a cost of \$1m - roughly one dollar for every word annotated. The structures of the PTB were loosely based on Chomsky's (1965) Extended Standard Theory (EST), which was considerably simpler and less abstract than contemporary Minimalist theory. Manually constructing a Minimalist treebank would therefore invariably have costed even more time and money, especially since no wide-coverage MG parser (or grammar) existed at the start of this project for proposing initial candidate trees for each sentence.

Deepbank (Flickinger et al., 2012) is another manually constructed treebank which uses the HPSG formalism and covers the same sentences as those in the PTB. As its name suggests, Deepbank's analyses are linguistically much richer and capture many more linguistic generalisations than their PTB equivalents. It is little surprise, then, that Deepbank took even longer than the PTB to complete (around 5 years).

Given the time and funding constraints of the current project, and the specialist knowledge of both Minimalist theory and the MG formalism that would be required by a team of annotators, a more promising approach to treebank creation seemed to be to convert an existing treebank, such as the PTB, into an MG treebank semi-automatically. There are two general approaches to such conversion that have been pursued for other formalisms, and these are discussed in the next two sections.

## 5.3 Semi-automatic conversion of an existing treebank into a different formalism

### 5.3.1 The transduction approach

The first approach to mapping a treebank in one formalism into a treebank in another formalism involves writing a transduction algorithm to transform the tree structures directly. This was the approach taken for the widely-used CCGbank (Hockenmaier and

Steedman, 2007), whose derivation trees were transduced from the PTB phrase structures. The conversion procedure was broken into two phases: the first was to map the existing trees into the general representational format of CCG. This involved binarising and lexicalising the source trees, making the complement/adjunct distinction explicit (using heuristics from Magerman (1994) and Collins (1999)) and then assigning CCG slash categories to terminals and non-terminals (using the traces of the PTB to determine where type raising and composition operations should apply). The second phase involved non-trivially modifying and/or enriching the underlying phrase structures, either because the CCG formalism required this, or because the researchers disagreed with certain theoretical decisions made by the PTB's annotators. For example, many small clauses in the PTB were replaced by a two-complement analysis in CCGbank following Steedman (1996).

While the first of these two tasks involved a relatively simple and elegant transduction algorithm, the second task required coding up a great many ad hoc tree surgical operations. As the number of such operations increases, their complex interactions can become difficult to manage (which will make them increasingly difficult for future researchers to modify). This sort of approach is therefore arguably best suited to cases where the required changes to the underlying tree structures are, relatively speaking, not extensive. For example, although as noted CCGbank does make a number of changes to the underlying phrase structures of the PTB, it does not, for instance, introduce head movements, covert movements, reflexive/reciprocal anaphor and floating quantifier binding, null heads, shell and X' phrase structures, movement of AGENT subjects from spec-vP to spec-TP, and so on.

A further issue is that given the many competing analyses for any given construction in the Minimalist literature, no single MG treebank will be universally accepted, and so it was desirable to make the structures of the treebank as easily modifiable and extendable as possible. Tree transduction methods are arguably not ideal in this regard, as modifying treebanks produced in this way involves either modifying the original transduction code, which as noted may be very complex owing to the cascades of tree surgical procedures, or writing additional transduction code.

A number of authors have proposed algorithms for transducing PTB-style trees into LTAG trees (Neumann, 1998; Xia et al., 2000; Chen et al., 2006; Demberg and Keller, 2008) by first decomposing them into elementary trees and then computing the LTAG derivation tree from the manner and order in which those elementary trees are recombined to produce the LTAG phrase structure tree. All of these methods involve first

identifying the trunks of *initial trees* (i.e. the path from the root of the initial tree to its lexical anchor/head) using a head-percolation table. Using similar heuristics to those proposed in Collins (1999) and Magerman (1994), Chen et al. (2006), Xia et al. (2000) and Demberg and Keller (2008) also distinguish between the complement and adjunct nodes which are dominated by nodes appearing along the trunk; the complement nodes are retained as substitution nodes on the trunk, while the trees they dominate become separate initial trees; the adjunct nodes, meanwhile, are factored out as separate *auxiliary trees*, which are later re-adjoined in binary branching fashion to the trunks they were extracted from (complements are reinserted into their substitution nodes).

In addition to basic modificational adjuncts, LTAG localises long-distance dependencies by assuming the existence of predicative adjuncts. For instance, the sentence *what did she say Pete eats for breakfast?*, would be generated from an initial tree whose string yield is *what Pete eats for breakfast* and an auxiliary tree with the yield *did she say* which is adjoined into a position between *what* and *Pete* in the initial tree. These types of adjuncts are factored into separate auxiliary trees by identifying complements of the initial tree which share a category with some node along the trunk of that tree (in this case an S node). The null co-indexed categories representing the traces that mark these long distance dependencies in the PTB can simply be retained intact in the LTAG initial trees.

A method for transducing PTB structures into HPSG trees is presented for English in Miyao (2006) and Miyao et al. (2005), and for Chinese in Yu et al. (2010) who use the Chinese Penn Treebank (Xue et al., 2005) as the source corpus. Under this approach, the structures of the source treebank are first annotated semi-automatically to be *partially-specified derivation trees*, which are trees annotated with HPSG *schema* names (schemas being abstract rules) and some of the features of the HPSG *signs* (signs being feature-value matrices used to label words and phrases and expressing various constraints and dependencies between them). As was the case for CCGbank, this stage requires some tree surgical procedures to be performed on the source trees in order to make them better fit with the target formalism's analyses. For example, Yu et al. (2010) altered the analysis for relative clauses in the Chinese Penn Treebank to remove the null *wh*-operator heads. They also designed 49 rules (following Hockenmaier and Steedman (2007)) to correct certain inconsistencies in the underlying annotation. A further 48 rules (following Miyao (2006)) were also created to automatically construct the derivation tree annotation, for example by using pattern matching to assign certain constituents to the correct HPSG *schema*. Finally, the schemas and principles of the

HPSG grammar are applied to the partially-specified trees to check their consistency and also to fill out any unspecified constraints and features.

All of the aforementioned transduction approaches share in common the fact that no new long-range dependencies are introduced. This contrasts with one of the primary goals of the MGbank project, which was to introduce several additional movement dependencies which feature prominently in Minimalist theory, including polarity item licensing, head movement and reflexive/reciprocal anaphor binding.

### 5.3.2 The parser approach

An alternative approach to treebank conversion which has been pursued for various formalisms (e.g. by Cramer and Zhang (2010) for HPSG, and Stabler et al. (unpublished research) for MCFG), is to use a parser to generate the target treebank. Cramer and Zhang (2010) equip their parser with the grammar of Cramer and Zhang (2009), which consists of a hand-crafted core grammar and an extended grammar which was extracted automatically from the Tiger treebank (Brants et al., 2002) which contains 50,000 sentences annotated with both phrase structures and dependencies. The parser was used to reparse the sentences of this treebank and any resulting HPSG trees which recovered all the original Tiger dependencies were retained in the new treebank. Cramer and Zhang (2010) note that their approach is very much precision-oriented in that it aims to produce highly detailed analyses of German constructions while rejecting ungrammatical sentences as much as possible, a goal which is shared by the current project. Cramer and Zhang (2010) also note that such precision comes at some cost to coverage, however, as their treebank only covers around 25,000 of the Tiger sentences (though these were only selected from the first 45,000 sentences of the Tiger treebank, so the overall coverage was 55.6%, which is very close to MGbank's current coverage of the PTB trees (55.7%)).

Stabler et al. (unpublished research) employed a similar approach to construct a Multiple Context Free Grammar (Seki et al. 1991; MCFG) treebank for English. Their MCFG parser was equipped with a grammar that had been painstakingly hand-crafted over the course of more than 20 years. This was used to reparse all the sentences of Ontonotes Weischedel et al. (2012), with the resulting trees being scored based on the number of constituencies from the original treebank which they successfully recovered; the highest scoring tree for each sentence was added to the treebank, provided it recovered a certain threshold number of constituencies. Those authors achieved an

impressive 99% coverage of the Ontonotes corpus using this method - a testament to the strength of their grammar.

An important advantage of the parsing approach to treebank adopted by these researchers is that it avoids the need for coding and modifying cascades of ad hoc transduction rules. Instead, the focus shifts to the design of the target grammar and a small set of heuristics for selecting the best tree from the set of candidates produced by the parser. The trees thus generated can be of any arbitrary level of complexity, and can differ substantially from those of the source trees, for example by containing many more long-distance dependencies and null heads.

Of course, a drawback of the parser-based approach is that it requires a pre-existing wide-coverage grammar for the target formalism, and as we have seen, no such wide-coverage grammar for MGs (or indeed MP) existed at the start of this project.

## 5.4 Autobank

In order to overcome this obstacle, a Minimalist grammar and treebank development environment called Autobank was developed<sup>1</sup>. The treebanking aspect of Autobank is similar to the Parsebanker system for LFG (Rosén et al., 2009), except that as well as supporting manual discrimination between candidate trees generated by the MG parser, Autobank also includes a module for performing such discrimination automatically once a small seed corpus has been constructed manually. This was important here given that MGbank was created by a single linguist in a relatively short time frame.

Autobank provides the linguist with a powerful graphical user interface enabling the creation of a sophisticated and globally consistent MG. Importantly, the development of the grammar is largely corpus-oriented: the annotator creates MG trees for existing PTB sentences. This allows the system to learn a set of labelled dependency mappings between the PTB trees and their MG equivalents, so that it can later take over and finish off the annotation automatically. This methodology also forces the researcher to confront the actual performance data which wide-coverage parsing systems must be able to handle, although there is also a facility for adding additional sentences to the corpus in order to cover certain linguistically interesting constructions (e.g. parasitic gaps, across the board (head) movements, *promise*-type subject control, etc) which are either infrequently represented in the PTB or absent from it altogether. Once enough trees have been annotated by hand, the automatic module can be used to

---

<sup>1</sup>The work presented in this section is based on work presented in Torr (2017)

annotate the remaining sentences of the corpus. Throughout the development process the researcher is able to see how their extensions and modifications in one area of the grammar affect other areas, allowing them to minimise both overgeneration and under-generation and therefore keep the grammar as efficient, generalisable and constrained as possible.

The various steps in the treebank conversion process are shown in fig 5.1. The PTB first undergoes an initial preprocessing phase. Next, the grammar engineer uses Autobank's GUI to manually construct a seed set of MG trees by annotating lexical items on PTB trees with MG categories and then selecting from among a set of candidate parses which are output using these categories by a non-statistical MG parser. This parser is a reimplementation of Harkema's (2001) CKY parser for MGs, but it has been adapted to use the Extended Directional Minimalist Grammar formalism presented in the previous chapter. The operations of the CKY MG parser (which formed the basis for the wide-coverage A\* MG parser) are described in section 6.3 of the next chapter.

Once enough trees have been annotated by hand, the researcher can initiate Autobank's automatic annotation module, at which point the system extracts various dependency/constituency mappings between the PTB source trees and the manually created target trees in the seed set, in order to compute the dependency-based scoring function. A statistical supertagger is also trained on the hand-crafted seed set so that the lexical annotation can now be performed automatically. For each unannotated sentence, the supertagger then assigns a set of candidate categories to each of the words in that sentence, and the parser uses these categories to generate a set of candidate parses for that sentence. The system then scores these candidate parses (using the dependency-based scores and various other heuristics, including a constituency-based comparison, for breaking any ties) and adds the one with the highest score to MGbank.

These processes are discussed in detail in the following three sections.

### **5.4.1 Preprocessing**

Autobank includes a module for preprocessing the PTB which corrects certain mistakes and incorporates some additional annotation that was done for the PTB following its initial release. This includes the additional NP structure from Vadas and Curran (2007) and the additional structure and role labels for coordination phrases recently released by Fidler and Goldberg (2016). The reason for adding in this structure was to help the automatic generator to choose the best tree from the set of candidates pro-

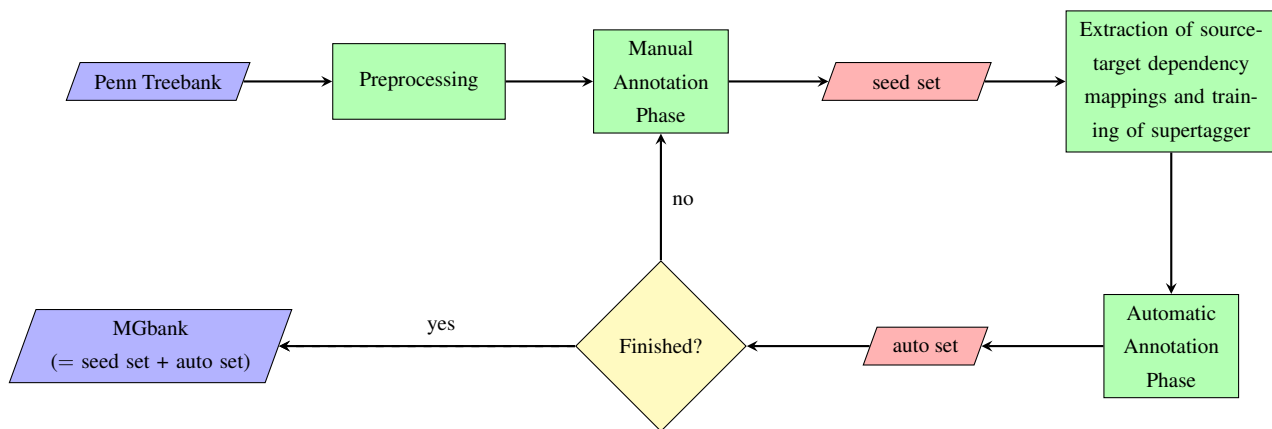


Figure 5.1: Autobank architecture.

duced by the parser during automatic annotation. For example, in the original PTB, the phrase ‘stock exchange collateral’ has a flat ternary branching structure, whereas Vadas and Curran rebracket it as [[stock exchange] collateral]. During automatic generation, the MG parser will not produce the flat ternary structure, because MGs are strictly binary branching, but it will produce both the correct [[stock exchange] collateral] and the incorrect [stock [exchange collateral]]. Because of the additional structure that was added into the PTB, the trees with the incorrect bracketing will be penalised on both the dependency-based and constituency-based scoring phases.

The structure for hyphenated compounds included in the Ontonotes 5 (Weischedel et al., 2012) version of the PTB was also included. The reason for doing this was to improve the quality of the final MG lexicon and avoid some data sparsity issues. For example, the original PTB contains the compound *seven-yen* as a single terminal, whereas the Ontonotes version breaks this up into three items, *seven*, *-* and *yen*. It is very unlikely that *seven-yen* will ever be encountered again by the parser, hence including this word in the lexicon is almost certainly not worthwhile. Furthermore, because such items are so rare, they will not contribute usefully to the statistical modelling. On the other hand, the parser is much more likely to encounter the three components *seven*, *-* and *yen* of this hyphenated word in the future, and so breaking these up and including them as separate entries in the lexicon does make sense, and they will also now contribute more usefully to the statistical modelling.

The semantic role labels of PropBank (Palmer et al., 2005) and Nombank (Meyers et al., 2004) have also been added onto PTB non-terminals,<sup>2</sup> along with the governing

<sup>2</sup>Among other things, these crucially distinguish adjuncts from arguments, raising/ECM from subject/object control, and *promise*-type subject control from object control/ECM.



word and its span (see fig 5.7). These will be useful when we come to look at the extraction of the dependency mappings which are the primary means by which candidate trees are scored during the automatic generation. PropBank also includes additional antecedent-trace co-indexing over and above what was included in the original PTB, which we have also imported, and some of this implies the need for additional NP structure beyond what Vadas and Curran have provided. For instance, in the phrase, *the unit of New York-based Lowes Corp that \*T\* makes kent cigarettes*, the original annotation has *the unit* and *of New York-based Loews Corp* as separate sister NP and PP constituents (with an SBAR node sister to both), both of which are co-indexed with the subject trace (*\*T\**) position in PropBank. In such cases an additional NP node was added resolving the two constituents into a single antecedent NP.

A number of corrections and amendments were also made to the PTB's tag set. The reason for this was that these tags are used, in conjunction with the CCGbank lexical categories (see below), as input tags for the statistical supertagger that was used during the automatic annotation phase. Thus, the more accurate and rich the information in each PTB preterminal category, the more accurate the supertagger is likely to be. In many cases these modifications were made following a process of trial and error in which the output of the automatic generator was inspected for errors and, where the fault lay with the supertagging model, it was sometimes possible to correct this by adding additional information onto the PTB preterminal tags.

For example, the PTB uses a generic PRP preterminal category for all personal and reflexive pronouns. This meant that the MG supertagger was sometimes mistagging *ourselves*, for example, either with the wrong reflexive category (e.g. the tag for *himself*) or a personal pronoun category. To correct this, suffixes were added to the PTB category in order to provide the supertagger with additional information at the tag level. In the case of reflexives, for example, person, number and reflexive-marking suffixes were added, so that instead of having the tag PRP, *ourselves* now has the tag PRP1PLSELF. It may seem as though there is already sufficient information at the word level here for the tagger to make the correct decision. While this is true in principle, recall that the supertagger is (initially at least) only trained on the very small hand-crafted seed set of MG trees, and there is not always enough evidence in this seed corpus to allow the parser to make the right decision in all cases, particularly for relatively infrequent words like *ourselves* which share the same PTB tag as very frequent items like *it* and *he*, and hence can end up being tagged with the MG category for those other items because the supertagger gives too much weight to the source tag.

By providing additional evidence at the tag level we split the PTB tags up, thereby ensuring that these mistakes are avoided. The various corrections and modifications that were made to the PTB tag set are listed below, and the full updated PTB tag set is given in table 5.1.

- Following Hockenmaier and Steedman (2002a), we have corrected cases where verbs were incorrectly labelled with the past tense tag VBD instead of the past participle tag VBN.
- Person suffixes (1, 2, 3) were added to personal, possessive and reflexive pronouns. Number suffixes (SG or PL) were also added to all of these items, as well as to demonstrative pronouns and determiners. For example, *we* is labelled PRP1PL, *our* is labelled PRP\$1PL, and *that* is labelled DTSG.
- Gender suffixes (M/F) were also added to masculine and feminine pronouns. The tag for *his*, for example, is PRP\$M3SG. Additionally, gender was also added to proper and common nouns. For proper nouns (i.e. names), the NLTK names corpus was used, as this divides names into male and female categories. Where a name appeared in both the male and female category lists (e.g. *George*), both M and F were added. For neuter names (e.g. the name of a company), no suffix was added. For common nouns, a list of masculine and feminine nouns was created by hand (by combining various such lists from the internet). For example, feminine nouns include *woman*, *girl*, *actress*, *queen*, *policewoman* etc. Gender is used in the MGBank grammar to block analyses such as *\*Jack loves herself* and *\*the woman who loves himself*.
- SELF suffixes were added to reflexive pronouns. For example, *herself* is labelled PRPF1PLSELF.
- The proper noun NNP tag on country adjectives (*English*, *Japanese*, *American* etc) was replaced by the adjective tag JJ.
- The RB tag on negation particles (*not*, *n't*) was replaced by a NEG tag. This was to ensure that the MG category for the negative particle was assigned in such cases, rather than the category for an adverb.
- All number and date terminals (e.g. *1966*, *1.4*, *1/5/1975*) were replaced with a single word, *num*.

tag	description	tag	description
CC	coordinating conjunction	PRP1SGSELF	1SG reflexive pronoun
CD	cardinal number	PRP1SG	1SG personal pronoun
DT	determiner	PRP2	2SG/2PL personal pronoun
DTPL	plural demonstrative	PRP2PLSELF	2PL reflexive pronoun
DTSG	singular demonstrative	PRP2SGSELF	2SG reflexive pronoun
EX	existential <i>there</i>	PRP3PL	3PL personal pronoun
FW	foreign word	PRP3PLSELF	3PL reflexive pronoun
HYPH	(Ontonotes) hyphen	PRP3SG	3SG neuter personal pronoun
IN	preposition, subordinating conjunction	PRP3SGSELF	3SG neuter reflexive pronoun
JJ	adjective	PRPF3SG	3SG feminine personal pronoun
JJR	comparative adjective	PRPF3SGSELF	3SG feminine reflexive pronoun
JJS	superlative adjective	PRPM3SG	3SG masculine personal pronoun
LRB	opening brackets	PRPM3SGSELF	3SG masculine reflexive pronoun
LS	list marker	RB	adverb
MD	modal	RBR	comparative adverb
NEG	negative particle	RBS	superlative adverb
NML	nominal	RP	particle
NN	noun, singular or mass	RRB	closing brackets
NNF	noun, singular feminine	SYM	symbol
NNM	noun, singular masculine	TO	infinitival <i>to</i>
NNMF	noun, singular masculine/feminine	UH	interjection
NNP	proper noun, singular	VB	verb, base form
NNPF	proper noun, singular feminine	VBD	verb, past tense
NNPM	proper noun, singular masculine	VBG	verb, gerund or present participle
NNPMF	proper noun, singular masculine/feminine	VBN	verb, past participle
NNPS	proper noun, plural	VBP	verb, non-3SG present
NNS	noun, plural	VBZ	verb, 3SG present
PDT	predeterminer	WDT	wh-determiner
POS	possessive ending	WP	wh-pronoun
PRP	generic personal pronoun ( <i>one</i> )	WP\$	possessive wh-pronoun
PRP\$1PL	1PL possessive pronoun	WRB	wh-averb
PRP\$1SG	1SG possessive pronoun	“	opening quotation marks
PRP\$2	2SG/2PL possessive pronoun	#	#
PRP\$3PL	3PL possessive pronoun	\$	currency symbol
PRP\$3SG	3SG possessive pronoun	”	quotation marks
PRP\$F3SG	3SG possessive personal pronoun	,	comma
PRP\$M3SG	3SG masculine possessive pronoun	;; : _ ...	punctuation
PRP1PL	1PL personal pronoun		empty tag
PRP1PLSELF	1PL reflexive pronoun		

Table 5.1: The updated PTB tag set used in Autobank.

As discussed in detail below, the supertagger that is used in order to assign lexical categories to words during automatic generation takes as input the word, its PTB tag, and an enriched version of the CCGbank preterminal category for that word (which includes the corrected PTB terminal categories from Hockenmaier and Steedman (2002a)). Some preprocessing was therefore also carried out on the CCGbank trees in order to enhance the information available to the supertagger further. The changes that were made are given below.

- The Ontonotes structure for hyphenated compounds was grafted onto the CCG trees in order to ensure that the CCGbank, PTB and MGBank trees all had the same number of overt terminals. This was necessary for the supertagging to work successfully.
- The atomic conj category used for coordinating conjunctions in CCGbank was replaced by a full slash category indicating the category of the conjunct arguments and the final result. Examples of these categories are  $\{CCNP\backslash NN\}/NN$ ,  $\{CCADJP\backslash JJ\}/JJ$  and  $\{CCS\backslash S\}/S$ . This more articulated tag set for coordinators greatly enhances the supertagger's accuracy for these items during automatic annotation.
- In CCGbank, preterminals consist of the CCG category appended with both the original PTB category and Julia Hockenmaier's corrected version, which is often identical. An example of such a tag is  $N/N\_JJ\_JJ$  where the three components are separated by underscores. During preprocessing, the original PTB category is replaced by the version that was created for this project (which is in some cases also identical). So PRP would be replaced by 3SGMSELF if the terminal is *himself*. The CCGbank preterminal would then be  $NP\_PRP\_PRPM3SGSELF$ . Julia Hockenmaier made a great many corrections to the PTB preterminals, and so retaining her version allows the supertagger to exploit her improvements.
- For nouns, prepositions, subordinating conjunctions and adverbs, if a function tag was included on the highest non-terminal headed by that word, then this function tag was also included on the CCGbank preterminal. So for example, the tag  $\{\{S\backslash NP\}\{S\backslash NP\}\}/NP\_IN\_IN\_TMP$  is the preterminal for a temporal adverbial preposition, whereas  $\{\{S\backslash NP\}\{S\backslash NP\}\}/NP\_IN\_IN\_MNR$  is used for manner prepositions. This helps the supertagger to select the correct MG preposition category.

- Where a verb has arguments which are annotated as ARG0 (for agents) or ARG1 (for patients) in Propbank, these tags are grafted onto the CCGbank preterminal for the verb. This enables the supertagger to distinguish between verbs which take agent arguments, such as control verbs (*want to*, *try to*) and unergative intransitives (*laugh*, *cough*), and superficially similar verbs which do not, such as raising verbs (*seem to*, *appear to*) and unaccusative intransitives (*remain*, *arrive*). In CCG syntax, control and raising verbs have the same category, as do unaccusatives and unergatives (the interpretive differences must be encoded in a separate semantic layer of representation). In Minimalism, however, the semantics must be directly encoded in the syntactic structure, and so these different types of verbs have different syntactic categories. The additional Propbank annotation allows the supertagger to select the appropriate MG category. Where the ARG1 argument is a clause, however, an ARGS tag is added instead of an ARG1 tag. The reason for doing this was to allow the supertagger to also distinguish between object control (*he persuaded Mary to help*) and exceptional case marking (*he expected Mary to help*) constructions, as these also have the same category as one another in CCGbank (but not MGbank), but are generally distinguished in Propbank by the fact that the clausal argument of an ECM verb is marked ARG1 whereas that of an object control verb is marked ARG2, with its DP argument being marked ARG1.

### 5.4.2 The manual annotation phase

Autobank provides a powerful graphical user interface for developing a wide coverage MG. The development is corpus-oriented, in that the user is presented with PTB trees and required to construct equivalent MG trees for them. This is achieved by relabelling the overt preterminals of the PTB tree with MG lexical categories, sending these to the parser, and then choosing the correct parse from the set of candidate analyses that are returned.

The main annotation environment is shown in fig 5.2. The PTB tree and its MG candidates are respectively displayed in the top and bottom window. Between these are a number of buttons allowing for easy navigation through the PTB, including a regular expression search facility for locating specific construction types by searching both the PTB/CCGbank bracketings and the string. The user can also choose to focus on sentences of a given string length. On the left, the sentence is displayed from top to

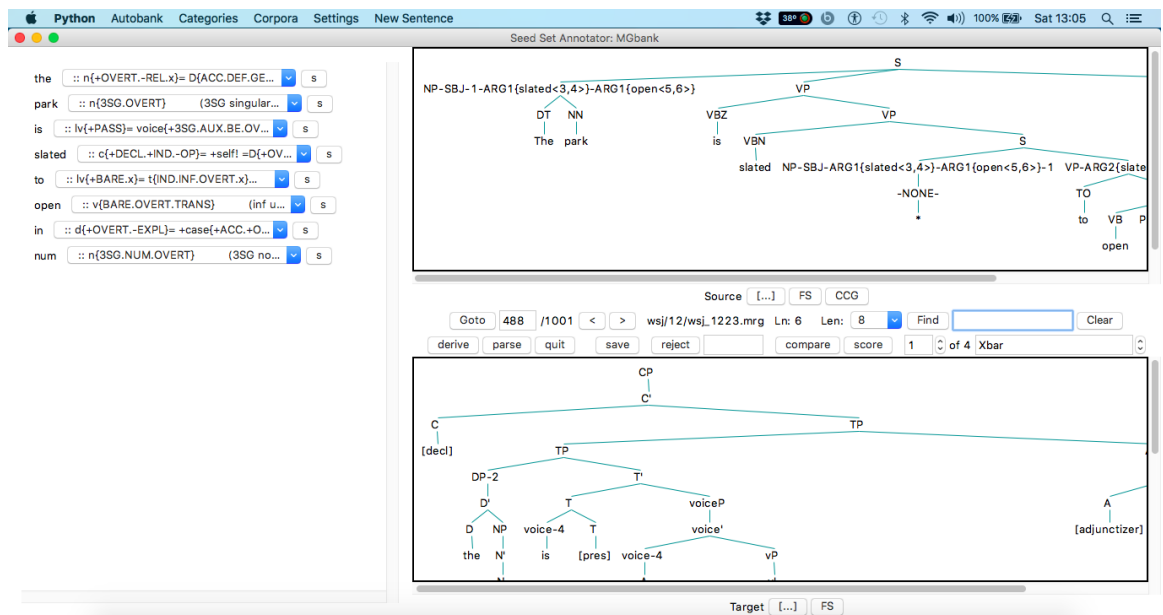


Figure 5.2: The main annotation environment.

bottom, each word with a drop-down menu listing all MG categories so far associated with that word's PTB preterminal category.

Both overt and null MG lexical categories are added to the system using the interfaces shown in figs 5.3 and 5.4 respectively, which are also used to subsequently modify lexical categories. Whenever the user attempts such a modification, the system will reparse any sentences in (both the manually and automatically created portions of) the treebank containing the category in question to ensure that the Xbar trees which are saved for them can still be generated following the modification.<sup>3</sup> This ensures that the grammar remains consistent, as the engineer is unable to modify a category to accommodate one construction if that modification triggers undergeneration or incorrect generation with respect to other constructions. Instead, they will need to either create an entirely new category for the new construction, or remove the trees that block the modification from the seed set and then re-add them with a different analysis.

Once the engineer has selected an MG category for each word in the sentence,

<sup>3</sup>The *derivation* tree will of course always change as this contains the features being modified. As long as the geometry of the *Xbar* tree remains the same, however, the predicate-argument structure will be preserved, regardless of the features that gave rise to that geometry. It is usually only the fine-grained selectional and agreement features which can be edited without affecting the geometry of the Xbar tree. Selectors, selectees, licensors and licensees are all structure building features, hence they give rise to different phrase structures when they are modified. An exception to this is the suicidal licensors, which where they do not check any feature do not lead to the creation of structure in the Xbar tree, and hence can be freely added or removed in many cases.

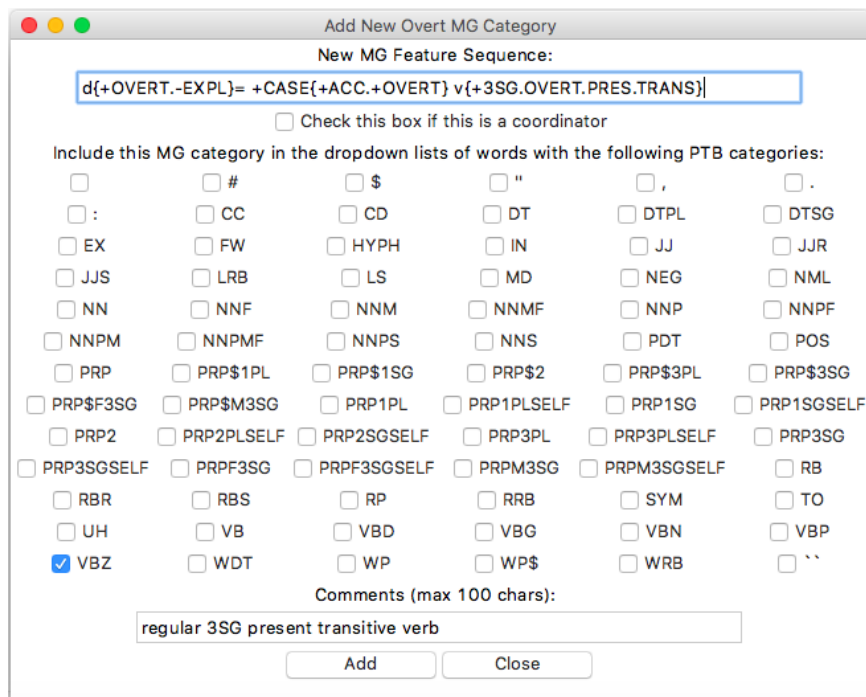


Figure 5.3: Overt category entry window.

clicking `parse` causes the parser to return all possible parses using these categories. Once returned, the trees can be viewed in several formats, including multiple MG derivation tree formats, and Xbar tree and MG (bare phrase structure) derived tree formats. Pairs of candidates can be viewed simultaneously for comparison (see fig 5.5), and there is the option to perform a diff on the bracketings, and to eliminate all trees containing an incorrect substructure from consideration with a single click. Once the user has identified the correct tree, they can click `save` to save it to the seed set. All trees in MGBank (both manually and automatically generated) can easily be viewed and removed using the native file system, and there are facilities for searching the corpus for particular lexical categories/constructions, as well as for searching the lexical category set based on the features of the category or the comments which were added to its entry.

There will of course be occasions when the parser fails to return any parses because the grammar engineer has not properly thought through all the steps of the derivation. In these cases it is very useful to be able to build up the derivation manually step-by-step in order to identify the point where it fails, and Autobank provides an interface for doing this (see fig 6.3). Whereas in annotation mode null heads were kept hidden for simplicity, in derivation mode the entire null lexicon is available, along with the overt

Add New Covert MG Category

New MG Feature Sequence: 

Morpheme label (e.g. decl):

Check this box if this is a coordinator

Select which null lexicon to add this category to:

Covert  Extraposer

Type Raiser  Tough Operator

Comments (max 100 chars):

Figure 5.4: Null category entry window.

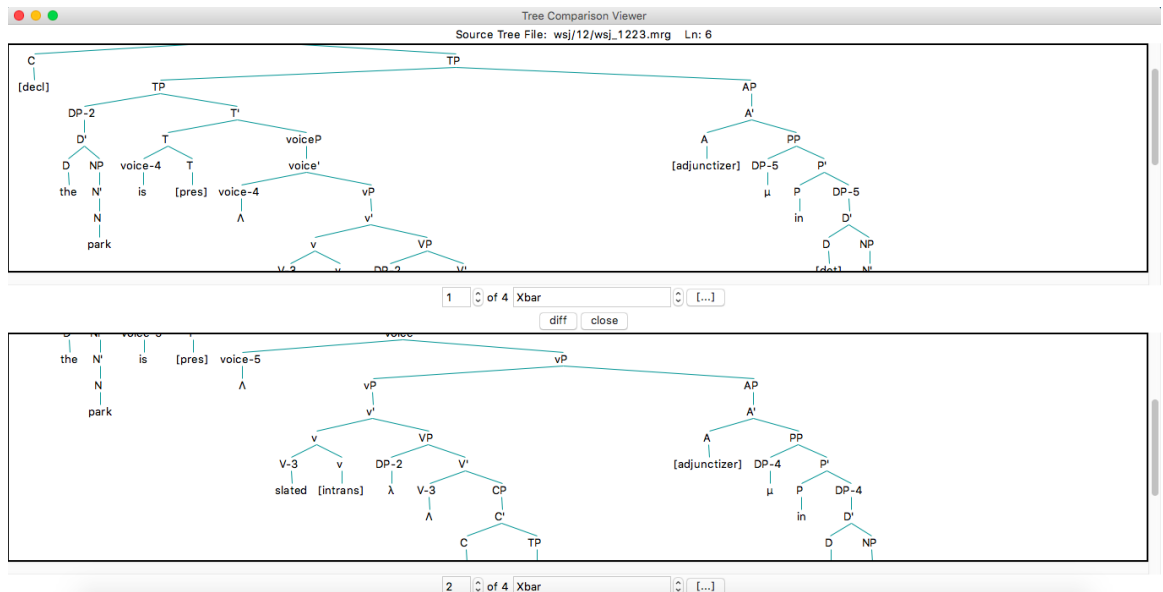


Figure 5.5: Tree comparison window.

categories the user selected on the main annotation screen.

A further useful feature of this system is that additional sentences can be added to the treebank and annotated. This is advantageous for two reasons. First, it allows constructions which are rare or absent entirely in the PTB to be included in MGbank. For example, there are no examples of *promise*-type subject control in the PTB, so some sentences featuring this construction were added; all sentences added in this way are stored in a separate folder from the PTB portion of MGbank. The second way in which this feature is useful is that it allows the user to test out their proposed analysis for a given construction type on a simpler and shorter version of an existing PTB sentence. Working with shorter sentences makes it much easier to identify where



the derivation is failing, but sometimes the only example of a construction that can be found is contained in a very long sentence. The ability to work on just the part of the sentence the grammar engineer is interested in, say just a single clause minus irrelevant modifiers, allows the grammar engineer to confirm whether a problem lies with their analysis of that construction itself or with some other part of the sentence.

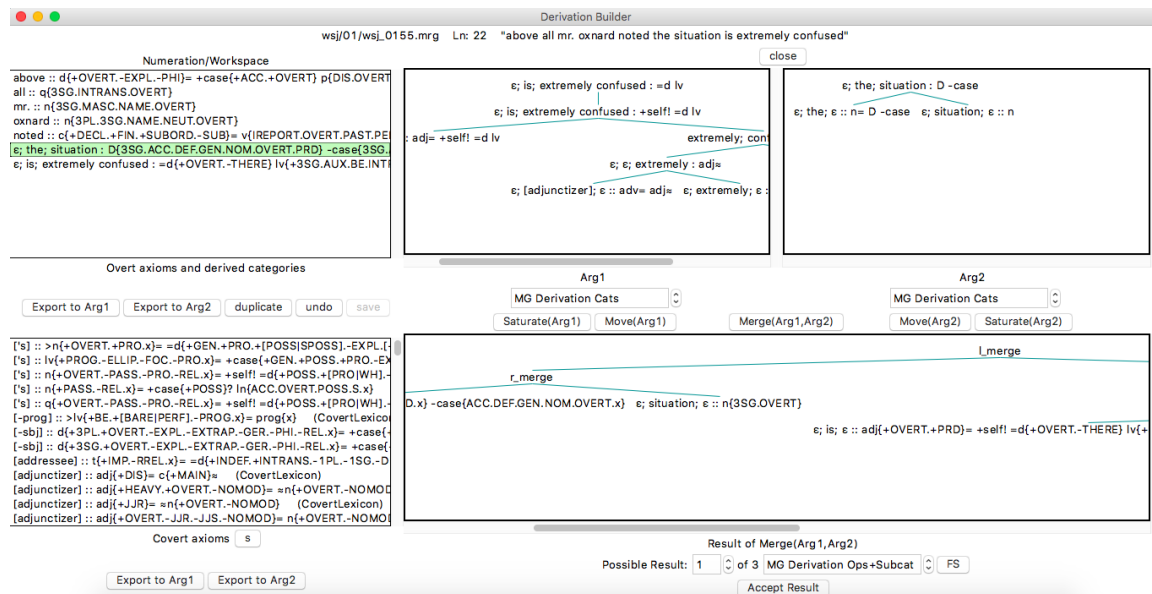


Figure 5.6: The step-by-step derivation builder.

### 5.4.3 The automatic annotation phase

Once the seed set has been created, the automatic annotator can be initiated from the command line. It is advisable to manually annotate at least 800 trees covering as wide a range of construction types as possible before running the automatic annotator, as it requires a certain amount of data to train the supertagger (see below) and must be equipped with a reasonably wide coverage grammar in order to work effectively.

During the manual annotation phase, the grammar engineer was responsible for annotating lexical items with MG categories and also for selecting the correct parse from the list of candidates outputted by the parser. The parsing step was already performed automatically, of course, except in cases where the derivation builder was used to construct the MG tree manually step-by-step. For this next phase, however, the lexical annotation and tree selection steps also need to be fully automated. Once the automatic generator is initiated for the first time, it therefore begins by training a statistical supertagger, which will be responsible for performing the lexical annotation.

Next, it extracts a set of dependency mappings between the seed trees and their PTB equivalents. These mappings will later be used, along with various other heuristics, to score the candidate trees and select the best one to be added to the MG treebank. In the following two subsections, we look at the automatic lexical annotation and scoring of candidates in detail.

The supertagger that was used for the manual annotation task was the maximum entropy C&C multi-tagger described in Clark and Curran (2010). This supertagger was originally designed for CCG supertagging, taking as input a word and a PTB pos tag (first generated by a separate pos-tagger), and outputting a CCG lexical category. This supertagger has since been outperformed by other neural network supertaggers (Lewis et al. (2016); Xu (2016)) which have the advantage that they do not rely on an error-prone pos tagger to provide the input tags (instead, they use word embeddings). For the treebank generation task, however, this is not a consideration as here the supertagger has access to the gold standard tags of the source treebank. Importantly, the C&C supertagger outputs all tags which fall within some factor  $\beta$  of the probability of the most probable tag. This parameter is set to 0.0001 within Autobank's code, but not all of the tags outputted by the supertagger are tried at once. Instead, Autobank initially tries the most probable tags outputted by the supertagger, and all tags within some factor  $\gamma$  of the best tag ( $\gamma$  can be set as a parameter when the automatic annotator is run), and only if it fails to build a parse with them will it try more tags by incrementally lowering  $\gamma$ . The ability to output multiple tags is very important for parsing, because with 1-best supertagging even a single mistake by the supertagger will result in either no parses or incorrect parses being generated.

The C&C supertagger was retrained to output overt MG categories instead of CCG categories. However, one problem was that this supertagger had to be trained on the very small set of around 800 manually constructed MG trees that have corresponding PTB trees. Unsurprisingly given the very small size of this training set, using just the PTB POS tags as the input tags to the C&C supertagger did not yield very good results. The problem is that these input tags are not very syntactically informative, and so a large amount of contextual data for each tag is required in order to build an accurate model. The tag VBZ, for instance, could refer to a great many different types of third person singular present tense verbs, including intransitive, transitive, ditransitive, prepositional dative, raising, reporting, ECM or object control verbs, etc.

The solution to this problem was to increase the amount of information available to the supertagger by providing it with much richer input tags. This was achieved

by training the supertagger using gold-standard CCGbank lexical categories (which included the PTB POS tags and were further enriched as described in section 5.4.1) as the input tags; these gold standard tags were then also provided to the supertagger at test time, i.e. during automatic annotation. This was in fact the reason for choosing the C&C supertagger here, as unlike more recent supertaggers it was designed to take auxiliary tags as input along with the words. Because CCG categories contain a great deal of information about the subcategorisation frame of a given word, they are much closer in granularity to MG categories than are the much coarser PTB POS tags. For example, in CCGbank, an intransitive verb has the category  $S \backslash NP$ , whereas a transitive verb has the category  $(S \backslash NP) / NP$  and a ditransitive has the category  $((S \backslash NP) / NP) / NP$ . These types of distinctions enabled the supertagger to perform well even though it was trained on so few sentences.<sup>4</sup>

#### 5.4.4 Automatic scoring of candidate trees

During automatic annotation, the system must choose which tree to add to the MG treebank from the set of candidates produced by the parser. This is achieved using a cascade of heuristic scoring metrics. As each metric is applied, all parses whose score is worse than that of the highest scoring parse for that metric are eliminated. If only one parse remains, the scoring is finished and this parse is added to the MG treebank, otherwise the process continues on to the next metric. The first and therefore most important metric that is applied is one which compares the dependencies between the PTB tree and each MG candidate Xbar tree. This metric is quite complex, and is described in full in section 4.7.2.1 below. The remaining metrics are discussed in section 4.7.2.2.

---

<sup>4</sup>Note that a feature was added to Autobank such that if the grammar engineer adds a sentence not in the PTB to the treebank whose words are however a subset of, and in the same order as, those of some PTB/CCGbank sentence, the system will detect this and reconstruct a simplified CCGbank tree for this sentence from the original larger CCGbank tree. For example, the simplified sentence: *A levelling off of farmer selling removed some of the downward pressure* would be matched with the larger sentence: *A levelling off of farmer selling tied to the harvest also removed some of the downward pressure on futures contract prices*. This allows the system to determine what the CCGbank preterminal categories should be for each of the words in the new sentence, and then use that sentence as part of the training data for the supertagger. The reason for doing this was that some constructions were contained inside very large PTB sentences which it was not practical to annotate manually.

#### 5.4.4.1 Dependency-based scoring

The basic idea here is to select the MG tree which is most similar to the PTB tree in terms of the dependencies it encodes. One point is awarded to each MG candidate for every dependency which it shares with the PTB tree, with only the highest scoring trees being retained. There are several types of dependencies of varying degrees of abstraction which are used to perform this evaluation. The first are simple unlabelled, undirected word-word dependencies. For example, if in both the PTB tree and the MG tree *yesterday* is analysed as being in a dependency relation with *helped* in *Pete said that Jack helped Mary yesterday*, then the MG candidate tree receives one point.

Binary dependencies are calculated for each non-unary PTB non-terminal by identifying its head and non-head daughters and then traversing down the tree to retrieve their head words. The identification of head children is achieved using two variants of Collins' (1999) head-finding rules (these are listed in full in appendix B), the first of which encodes syntactic dependencies, while the second encodes semantic dependencies. The main difference between the syntactic and semantic dependencies are that in the former the preposition is treated as the head of the PP, the complementizer (if present) as the head of Sbar, the auxiliary (if present) as the head of S and the determiner (if present) as the head of the NP, whereas in the latter the noun is always treated as the head of PP and NP and the verb as the head of S and Sbar. In addition, for coordinate structures, the syntactic head is the coordinator itself, whereas the heads of the conjuncts are treated as multiple semantic heads.

The rules specify the possible head daughters of a given non-terminal in order of preference. For example, the syntactic and semantic head daughter finding rules for PPs and NPs are given below.

##### (335) Syntactic head-finding rules

- a. PP: (IN, R), (PP, R), (TO, R), (NP, R), ([NN, NNM, NNF, NNMF, NNS], R), ([NNP, NNPM, NNPF, NNPMF, NML], L), (S, R), (VBG, R), (VBN, R), (RP, R), (FW, R)
- b. NP: (VP, R), ([DT, DTSG, DTPL], L), (QP, R), (NNS, R), ([NN, NNM, NNF, NNMF, NML], R), ([NNP, NNPM, NNPF, NNPMF], L), (NNPS, L), (NX, R), (NP, L)

##### (336) Semantic head finding rules

- a. PP: (PP, R), (NP, R), ([NN, NNM, NNF, NNMF, NNS, NNP, NNPM,

- NNPF, NNPMF, NML], R), (S, R), (TO, R), (VBG, R), (VBN, R), (IN, R), (RP, R), (FW, R)
- b. NP: (VP, R), (NNS, R), ([NN, NNM, NNF, NNMF, NML], R), ([JJ, NNP, NNPM, NNPF, NNPMF, NNPS], L), (NNPS, L), (NX, R), (NP, L), (QP, R), ([DT, DTSG, DTPL], L)

When looking for the syntactic head word of a PP, the system starts by looking at the first tuple in the list given in 335a, which is (IN, R). The first member of this tuple specifies the PTB category that should be searched for, while the second specifies the direction of the search through the daughters of the non-terminal whose head daughter is being located. Here R means that the system should start with the rightmost daughter of the PP and, if that is not of category IN, the system should then progress leftward through each of the other daughters, stopping as soon as it discovers a daughter of category IN. In a canonical PP such as *to the agent* in fig 5.7, it will discover an IN in the leftmost position and assign the terminal dominated by this IN (here *to*) as the syntactic head word. If no IN daughter were present, however, the system would move on to the next tuple (PP, R), which tells it to again start searching the PP's daughters from the right but this time to stop if and when it discovers a PP. The system will keep progressing through these tuples until it finds an appropriate head daughter. If the head daughter is not a preterminal category, the system carries on down the tree, treating this head daughter as the new mother node and again applying the head daughter finding rules beginning with the first tuple. This continues until a head word is discovered. Note that square brackets indicate that the system can select any of the categories contained within them as the head child with equal preference.

Phonetically null terminals are never selected as heads because they cannot be reliably matched on both the PTB and MG sides. Null complementizers are represented in the PTB as a -NONE- node dominating a 0 terminal. These are never selected as heads because -NONE- does not appear in any of the head-finding rules. Traces have a tripartite unary branching structure consisting of a category node, say NP, dominating a -NONE- which in turn dominates a \*T\* terminal. The top-level category node is generally annotated with an integer which will also appear on the overt antecedent of this node elsewhere in the tree; whenever the system encounters such a trace node, it will use the integer to locate the overt antecedent and continue its search for the head word inside that. In this way, both local and long-distance dependencies are extracted from the PTB tree.

The main difference between the syntactic head-finding rule for PPs in 335a and its semantic counterpart in 336a is that in the latter the IN and TO categories appear much later on in the list, after NP. This encodes the assumption that while the syntactic head of a PP is the preposition it immediately dominates, its semantic head is the noun which this preposition most closely c-commands. For the PP in fig 5.7, for example, this rule will cause the system to select the NP as the semantic head daughter, rather than the IN. Furthermore, in the semantic head finding rule for NPs in 336b, the NN appears before DT (in contrast to the syntactic head finding rule in 335b for NPs where this situation is reversed). This means that the noun *agent* will ultimately end up being selected as the semantic head word of this PP.

For every non-unary non-terminal in the PTB tree, and for every non-head daughter of that non-terminal, a syntactic dependency is extracted between the non-head daughter's head word and the head daughter's head word. Where the semantic heads of any of the daughters differ from their syntactic heads, additional dependencies will be extracted for these. For example, for the VP non-terminal in the tree in fig 5.7 the three non-head daughters (both syntactically and semantically) are NP, PP and ADVP, with syntactic head words *the*, *to* and *now*, respectively, while the head daughter is VB with the syntactic head word *give*. This VP node therefore defines the following syntactic dependencies: *the-give*, *to-give*, *now-give*. In addition, the semantic heads of the NP and the PP differ from their syntactic heads, the former being *money* and *agent*, respectively. The semantic head words of the VB head child and the ADVP, meanwhile, are the same as their syntactic heads. This VP node therefore defines two further dependencies *money-give* and *agent-give*.

Both syntactic and semantic dependencies are collected from the PTB trees and MG Xbar trees; the fact that the dependency is semantic or syntactic is not retained and any duplicate dependencies are discarded. The reason for using both syntactic and semantic dependencies on both sides is to maximise the chance of the system finding a match. This also allows a certain degree of flexibility, in that the MG tree does not have to stick rigidly to the PTB tree's constituencies. For example, the PTB and MG trees may differ in terms of the level at which various modifiers are attached; in the PTB-style tree in fig 5.7, the temporal adverb is attached at the VP level and *give the money to the agent now* is therefore a constituent, whereas in the MGbank-style Xbar tree in fig 5.8 it is attached to the tense phrase (TP) and *give the money to the agent now* is not a constituent. However, because TP is part of the *extended projection* (Grimshaw, 1991) headed by the lexical verb *give*, there is a sense in which *now* is still a *semantic*

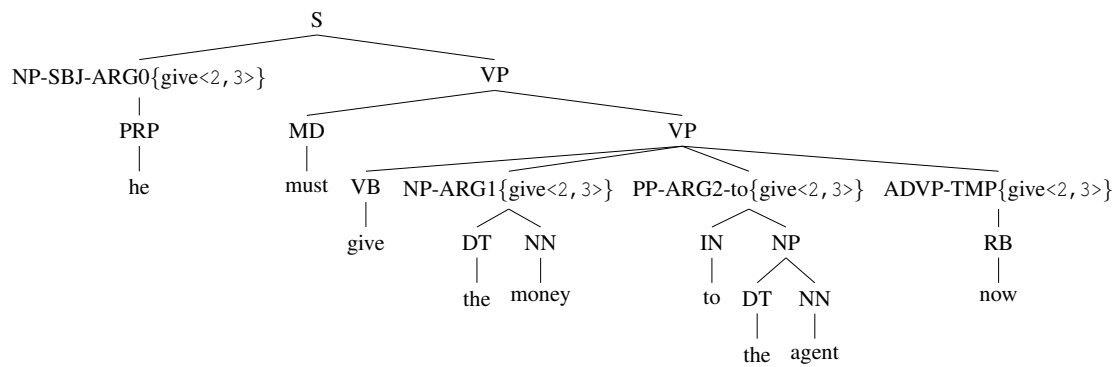


Figure 5.7: PTB-style tree for the sentence *he must give the money to the agent now*. The tree includes the Propbank labels and head spans which are added to non-terminals in the preprocessing step. For example, ARG0{give<2, 3>} specifies that the item this tag is attached to is an agent argument of a word *give* in span position [2,3].

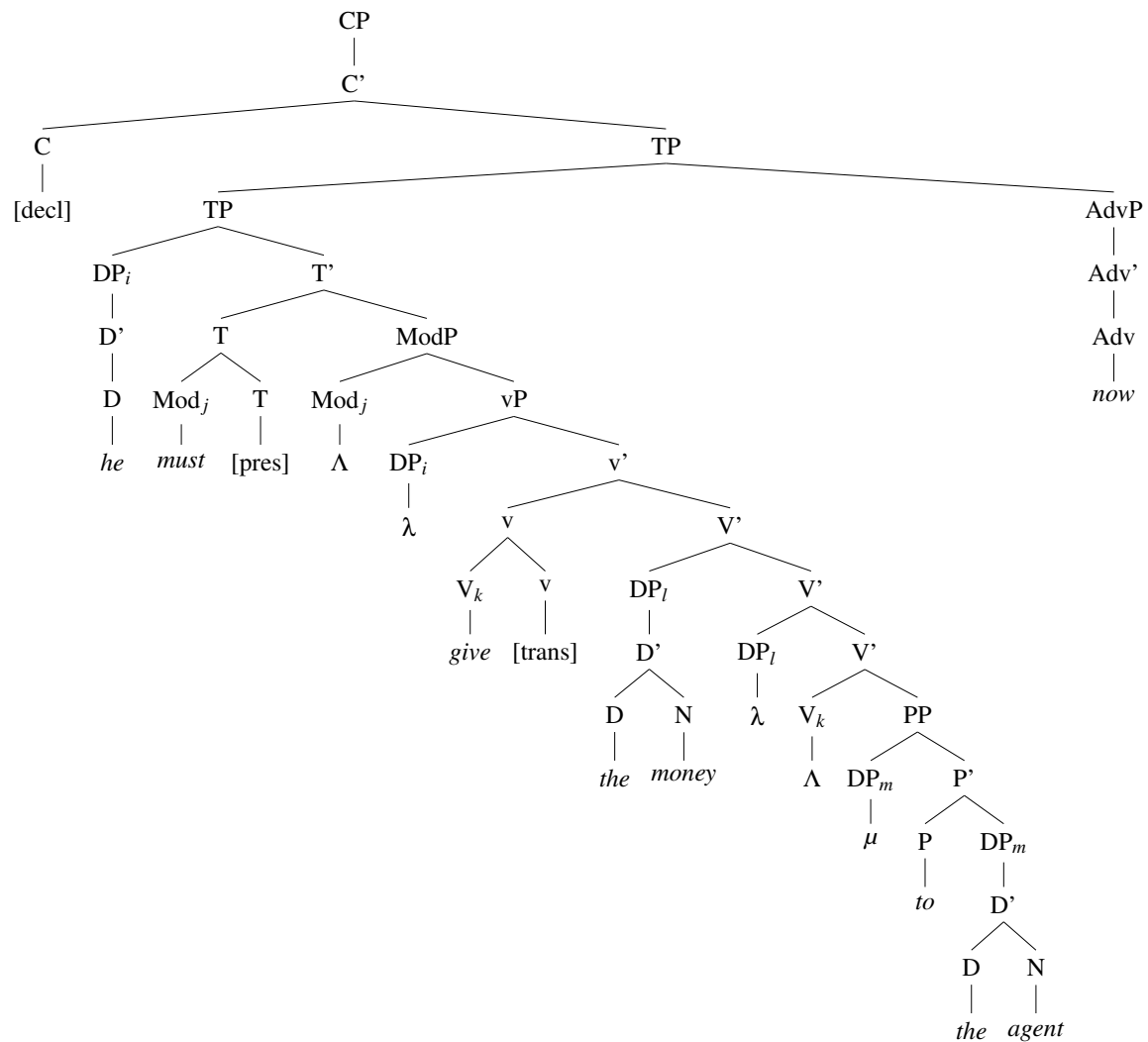


Figure 5.8: MGbank-style Xbar tree for the sentence *he must give the money to the agent now*.

dependent of this verb, despite being a *syntactic* dependent of the modal verb. The system therefore extracts a semantic dependency between *now* and *give* on the MG side, and this will be matched with the syntactic/semantic dependency between these two words on the PTB side, leading to a point being correctly awarded. This example illustrates why the dependency-based scoring is applied before the constituency-based scoring, as the latter would incorrectly penalise the MG tree here.

Unlike for the PTB trees, extracting dependencies from the MG Xbar trees does not entail the use of Collins-style head finding rules, as here the MG feature calculus of the corresponding MG derivation tree fully determines which daughter is the head and which the dependent: the head daughter is always the item which selected or licensed the other daughter, i.e. the one whose checked feature in the corresponding derivation tree was a selector or licenser. The only exception to this is that in the case of adjunct ( $\approx x/x \approx$ ) selector features it is the item with the selector which is regarded as the dependent.

As with the PTB trees, null heads in the MG Xbar tree are never selected as head words; where the head daughter is a phonetically null head, the system looks to the head of its complement and then to the head of the complement's complement, and so on, until it locates an overt word. Again as with the PTB trees, when traces are encountered, the system continues searching inside the overt antecedent of the trace. For example, in fig 5.8, the syntactic head word of the vP node is the null [trans] head, while the syntactic head of its DP dependent is the trace that is coreferenced with *he* in spec-TP. The null [trans] head does not appear in the PTB tree, hence there is no way any dependency that includes it on the MG side can ever be matched with a PTB dependency. Because [trans] is a null head, the system will look instead to the head of its complement, which is the V trace that is co-indexed with the moved lexical verb *give*. The system will therefore extract a syntactic *he-give* dependency<sup>5</sup> for this vP node, and will be able to match this with a corresponding dependency in the PTB tree. If no overt head is ever reached (as happens where the dependent is a null proform for example), or the head and dependent word turn out to be the same (as occurs for the *v'* node in fig 5.8), no dependency is extracted.

Semantic dependencies are also extracted from the Xbar trees in precisely the same way as syntactic ones, except that as well as continuing to search down the tree if a head daughter is a null lexical item, the system now also continues searching down the tree if the head word is not of category N, V, Adj or Adv, which are the four lexical

<sup>5</sup>It would also extract an identical semantic dependency, but would only retain one of these.



heads (i.e. the semantic hearts) of extended projections. For example, in fig 5.8, the syntactic head of C' will be the modal *must* which heads ModP. This is because both C and T are null heads, hence they do not qualify as head words for the purposes of dependency extraction. The semantic head word of C', however, will be the V head *give*, because it is the first head of category V, N, Adj or Adv that the system will encounter as it recursively searches the heads of each complement as it travels down the tree. The other difference between syntactic and semantic dependencies is that in coordinate structures the coordinator is treated as the syntactic head whereas the heads of the conjuncts are treated as being multiple semantic heads of the coordinate complex.

In addition to the simple unlabelled undirected dependencies, labelled directed dependencies were also extracted, similar to those used in Collins (1999). They include the head child and non-head child categories, the parent category, any function tags on the non-head child (present only in the PTB), and whether the non-head child appears to the left or right of the head child. Any tuples in the PTB tree which have the same head and non-head words (which can occur owing to syntactic movement operations) are gathered together into a *chain*<sup>6</sup> and the same is done for the tuples in the MG tree (we will also regard a single tuple as a one-membered chain). Mappings are then established between PTB chains and MG chains with the same head and non-head words. Before the automatic annotation begins, a set of these mappings is extracted from the hand-crafted seed set. Then, during automatic scoring, candidate trees are awarded a point for any chain mapping extracted from them which was previously encountered in the seed set. This allows the system to capture categorial and relational regularities in the mapping from PTB structures to MG structures.

In fig 5.7, the S node yields the following dependency tuple for the semantic head word of the non-head VP daughter: [VP, *give*, <2, 3>, NP, *he*, <0, 1>, S, [ARG0, SUBJ], left]. Following standard Minimalist assumptions, in fig 5.8, the agent subject is first generated in spec-vP before moving to spec-TP. There are therefore two non-terminals which define dependencies in which *he* is the dependent head word. The first is the vP, which defines the dependency tuple [v', *give*, <2, 3>, DP, *he*, <0, 1>, vP, left]], and the second is the TP, which defines two dependency tuples owing to the fact that its syntactic head word (*must*) differs from its semantic head word (*give*). These two tuples are [[T', *must*, <1, 2>, DP, *he*, <0, 1>, TP, left], and [[T', *give*, <2, 3>

---

<sup>6</sup>This term is used differently in this section to how it is used in chapter 3 (and in the rest of the thesis), where it refers to a mover inside an expression.

DP, *he*, <0, 1>, TP, left]. The vP tuple has the same spans for both its head word and non-head word as the second of the two TP tuples. We therefore group these two tuples into a chain, and define the following PTB-MG dependency chain mapping, where the tuples have been made more abstract by removing the words and their spans:

(337) [VP, NP, S, [ARG0, SUBJ], left] → [[T', DP, TP, left], [v', DP, vP, left]]

The two membered chain on the right hand side of the above mapping records the movement of the subject from spec-vP to spec-TP in the Minimalist tree, while the single membered chain on the left indicates that there is no corresponding movement in the PTB tree. The reason for removing the words and spans is so that a mapping extracted from a candidate tree can abstractly be matched with a mapping previously extracted from a tree in the seed set irrespective of the actual words and their span positions.

When the automatic annotator is first initiated, it will iterate over all of the Xbar trees in the seed set which have corresponding PTB trees, extracting a set of these mappings for all non-terminals it encounters. Each mapping is stored in three forms of varying degrees of abstraction. The first, most abstract mapping is of the kind given in 337; in the second, a lemmatised version of the head word is included, while in the third both the head and non-head words are included, again in their more abstract lemmatised forms. The two more reified versions of 337 are given in 338 and 339 below:

(338) [VP, *give*, NP, S, [ARG0, SUBJ], left] → [[T', *give*, DP, TP, left], [v', *give*, DP, vP, left]]

(339) [VP, *give*, NP, *he*, S, [ARG0, SUBJ], left] → [[T', *give*, DP, *he*, TP, left], [v', *give*, DP, *he*, vP, left]]

Including mappings of varying degrees of abstraction allows the system to recognise not just general phrase structural configurations, but also more idiosyncratic properties conditioned by specific lexical items, as in the case of idioms and light verb constructions, for instance. For example, the copula verb *be* is treated as a type of light verb in MGbank (its main purpose being to convert DPs, PPs and AdjPs into predicates). It therefore appears in the little v position in MGbank, rather than the main V position, and this is true regardless of the head word of the complement it takes. This fact is captured by the second type of dependency given in 338, which includes the head word but not the non-head word, and enables the system to favour automatically

generated trees which place *be* in *v*. On the other hand, the verb *have* in its non-auxiliary usage will generally appear in the *V* position, except where its complement is semantically headed by the word *bearing* (represented in the dependency tuples by the lexeme *bear*), in which case *have* is quite likely to be a light verb (as in *to have no bearing on a situation*) and must appear in the little *v* position. This fact is encoded by the third type of dependency mapping given in 339, as this includes both the head word and the non-head word.

During automatic scoring, a point is awarded to an MG tree for each instance of any one of these three types of mappings that is encountered *iff* that mapping was previously seen in the seed set. So if 337, 338 and 339 are all extracted for a new sentence to be annotated, but only 337 and 338 were seen in the seed set, 2 points will be awarded (in addition to the point that would be awarded for the basic unlabelled dependency matching). All trees with scores lower than that of the highest scoring tree are eliminated from consideration.

#### 5.4.4.2 Other metrics

The dependency-based scoring metric is the first and therefore the most important metric to be applied to the MG trees. It is often the case that this metric results in more than one highest scoring candidate, however. For this reason, there are a number of other metrics which are successively used to break ties. The other metrics are listed below in the order in which they are applied. All candidate trees which score lower than the highest scoring tree are eliminated at each stage. If only a single candidate remains after the application of a metric, this candidate is added to the MG treebank; otherwise, the next metric is applied to eliminate more candidates.

- Positive constituency-based evaluation - a point is awarded to each MG Xbar tree for every constituent span which it shares with the PTB tree.
- Negative constituency-based evaluation - a point is deducted from each MG Xbar tree for every constituent span which it contains which is not found in the PTB tree.
- Disprefer [pro-x] - a point is deducted for each instance of a null proform in the Xbar tree. Null proforms include null VP ellipsis heads ([pro-v]) and NOC PRO ([pro-d]).

- Disprefer [self] morphemes - a point is deducted for every instance of a null [self] head in the derivation.
- Retain trees with greatest lexical probabilities - the sum of the log probabilities assigned by the supertagger is taken and only the top scoring trees are retained.
- Disprefer traces - this metric deducts a point for every trace or instance of covert movement contained in a tree.
- Prefer right branching - Xbar trees with the greatest number of right-branching nodes are preferred.
- Prefer smaller trees - The Xbar trees with the fewest number of nodes are preferred.

#### **5.4.5 Constraining the parser's search space with PTB and CCG-bank constituencies**

One drawback of the parsing approach to treebank conversion presented so far is that the parser may propose completely bogus analyses which may be accepted into the new treebank as long as they are superior to all other analyses according to the scoring metrics. This could be because the grammar was not properly constrained or did not have the required coverage, or because the supertagger proposed incorrect categories which nevertheless led to one or more incorrect parses. A further problem was caused by the fact that the CKY MG parser attempts to construct the complete chart, often with multiple lexical categories per word, and without any proper statistical model of the derivation there was no way to prune the search space meaning that parsing became very slow on sentences of around 20 words or more.

In order to address these two issues, Autobank includes two features which use the constituencies of the PTB and those of CCGbank to constrain the hypothesis space of the parser as it builds up structures. Either or both of these features can be switched on or off during both manual and automatic annotation. During the creation of MGbank, for the automatic annotation phase (and in most instances of manual annotation), they were both left switched on in order to achieve the highest quality treebank possible.

The first feature enforces the rule that if a span is a constituent in the MG Xbar tree, then it should be a constituent in either the PTB or CCGbank. The reason for using CCGbank constituencies in addition to PTB ones, is that CCGbank's trees are,

like those of MG trees, binary branching. Hence including CCGbank trees results in the inclusion of a great many spans which will be present in the MG trees but absent from the PTB trees. This constituency-based rule is already much more permissive, therefore, than the positive and negative constituency-based scoring metrics discussed in the previous section, which only compare the MG trees to the PTB trees.

The second feature enforces the constraint that if a constituent undergoes movement, it should be an S, X or XP node in the PTB tree, where X can be any category such as N, V P etc. In this case the CCGbank tree spans are not considered. The idea here is to capture the linguistic constraint that only minimal or maximal projections can undergo movement, not non-constituents and not intermediate X' projections. This is part of a wider constraint within Xbar Theory that only maximal projections can be specifiers, complements or adjuncts (movement is always to a specifier position (or adjunct position for rightward movement in MGbank)). Because CCGbank trees are binary, many of the constituents in these trees correspond to X' nodes in the binary MG Xbar trees, hence we do not use them to define the spans of the sentence which can undergo movement.

When tested on the manually constructed seed set, these two constraints turned out to be somewhat too restrictive in certain instances. It was therefore necessary to relax them by adding additional spans to the set of allowable constituent spans. For example, in the PTB the PP dependents of nouns are always adjoined at the NP level whether they are adjuncts or complements, and CCGbank simply follows the PTB annotation. In MGbank, however, PP complements of nouns form a constituent with the noun itself to the exclusion of any determiners and/or nominal modifiers; it is only PP adjuncts which attach at the NP level. A span for the N+PP complex is therefore added to the set of PTB spans prior to parsing. Another span which is added relates to the fact that temporal adjuncts attach to the VP in the PTB tree but often attach to TP in the MG trees as we saw in the previous section. A span for the VP which excludes the span of the temporal adjunct is therefore added to the set of PTB spans, to prevent such derivations from being blocked. There are around 15 such rules in total, some of which are quite complex and are not discussed further here. They can all be found in the file `autobank.py` as individual functions with names like `add_n_pp_spans()` and `add_vp_minus_tmp_spans()`.

## 5.5 MGbank

The construction of a treebank which is relatively faithful to syntactic theory is challenging, and MGbank remains a work in progress. This section first provides some current corpus statistics and then presents an evaluation of the corpus in its current state. Despite the fact that the corpus is still under development, it has already been used to train and test the first ever wide-coverage MG parser, presented in the next chapter of this thesis.

### 5.5.1 Corpus statistics

The entire MGbank corpus currently consists of MG derivation, derived and Xbar trees for 27,701 sentences of between 1 and 50 words in length covering 463,065 word tokens (43.43%) of the PTB or 55.71% of its sentences. 1,128 of the MGbank trees were initially annotated by hand, and of those 290 are sentences which were added to the corpus rather than being present in the PTB. The other 26,573 trees were generated automatically using the method described in the previous sections. A further 100 trees from the automatically generated set were subsequently annotated by hand in order to construct a test set for evaluating the quality of MGbank (see the next section). These 100 are now included in MGbank as seed trees rather than automatically generated trees, meaning that there are in total 1,228 hand-annotated trees and 26,473 automatically generated trees in MGbank. In total, there are 1,127 MG lexical categories used in MGbank (all hand-crafted), of which 369 are null heads. The hand-crafted seed corpus contains 758 overt lexical categories, 576 (76%) of which made it into the automatically generated trees.<sup>7</sup> The MGbank lexicon consists of 47,634 items (including the 369 null heads), and the average sentence length is 16.9 words (vs 21.7 in the original PTB).

---

<sup>7</sup>The reason around a quarter of overt categories from the seed set did not make it into the automatically generated set is that there are a great many rare constructions in the Zipfian tail, and the seed set deliberately focused on annotating as many of these as possible. These constructions may simply not appear in the rest of the corpus, or even if they do, the one or two annotated examples of them in the hand annotated seed set may not have been enough to enable the supertagger to assign the relevant categories a high enough score during automatic annotation.

## 5.6 Evaluation of MGbank

As described in the previous few sections, MGbank was generated using a parser which was constrained using the structures of the Penn Treebank and CCGbank. The set of candidate trees was then scored using various heuristics. This process is imperfect and it is possible for incorrect trees to be added to the corpus during this automatic generation stage. For this reason, this section provides some evaluations to indicate the quality of MGbank in its current iteration. Section 5.6.1 first provides a global evaluation of the dependencies defined in the automatically generated trees vs those of a small hand-crafted test set of 100 trees. Section 5.6.3 then moves on to evaluating the corpus on specific constructions.

### 5.6.1 Global dependency evaluation

The automatically generated portion of MGbank was first evaluated on how well its trees recover the dependencies of a gold standard hand-annotated test set. One immediate problem was that all of the gold standard trees had been used as seeds to train the C&C supertagger as part of the treebank generation process, meaning that they could not also be used in the test set. A potential solution was to remove a portion of these trees and then regenerate them automatically by retraining the C&C supertagger on just the remaining hand-crafted trees and then reapplying the automatic generator to the trees which had been removed. However, fewer seed trees would likely have resulted in a degraded performance for the automatic generator, meaning that the evaluation probably would not have been a true representation of MGbank's quality.

An alternative approach which was therefore adopted was to hand-annotate a further 100 trees randomly selected from the automatically generated portion of the corpus (the annotation was performed blindly, i.e. without reference to the automatically generated structures), and then to compare the automatically generated trees with their newly hand-crafted counterparts. Sentences of less than 10 words are very easy to parse and for this reason were excluded from the test set; conversely, sentences of over 25 words in length can take a long time to annotate manually using Autobank and were therefore also excluded. The total number of words in this gold standard test set was 1689, meaning that the average sentence length was 16.9 words, precisely the same as for MGbank itself. The file names and line numbers of these 100 sentence are listed in Appendix D.

Table 5.2 gives the results for both syntactic and semantic (local and non-local)

	dep type	F1	P	R	E
syntax	LAB	84.30	84.39	84.20	17.00
	ULAB	94.73	94.84	94.63	24.00
	DULAB	93.58	93.69	93.48	24.00
semantics	LAB	80.07	79.02	81.15	17.00
	ULAB	93.15	91.92	94.40	36.00
	DULAB	90.87	89.68	92.10	36.00

Table 5.2: Local and non-local dependency recovery results for 100 trees generated automatically using Autobank, using hand-annotated counterparts as the gold standard. LAB = labelled, directed; ULAB = unlabelled, undirected; DULAB = directed, unlabelled; P = Precision; R = Recall; E = Exact Match.

dependencies and further breaks these down into three types of dependencies: 1. labelled, directed; 2. unlabelled, undirected; 3. unlabelled, directed. The extraction process was the same as that described in section 5.4.4.1 for the dependency-based scoring of candidate MG trees (except that chains were not used). For every binary non-terminal in the Xbar phrase structure tree, a dependency was extracted between the head word and the head of the non-head-child of that non-terminal. Where a trace node was encountered, the system would follow the chain of co-indices until it found an overt antecedent, meaning that both local dependencies and non-local dependencies created by all the various types of movement in MGbank are captured. For this reason, the dependency evaluation used here is considerably tougher than the purely local dependency evaluation used in Collins (1999), and is more akin to the third method used in Clark and Hockenmaier (2002).

As was described in section 5.4.4.1, the difference between the semantic and syntactic dependencies is that the syntactic ones are defined purely in terms of the MG feature calculus, meaning that the determiner is the head of the DP, the complementizer is the head of the CP, the preposition the head of the PP and so on, whereas for the semantic dependencies it is the heart of the extended projection which is defined as the head; thus, in semantic terms, the noun is the head of the DP and PP, and the main lexical verb is the head of the CP. In addition, the coordinator is the syntactic head of a coordinate phrase, whereas its conjuncts are (multiple) semantic heads.

The labels of the labelled directed dependencies were composed of the parent category along with the categories of the head child and non-head child and an indication of



whether the non-head child was the left or right daughter (as in Collins (1999)). Thus the label for the subject dependency in spec-TP would be TP-T'-DP-left. Because non-local dependencies are also captured, subjects of active transitive verbs and unergative intransitive verbs would also have a dependency for their base-generated position in spec-vP, labelled with vP-v'-DP-left, whereas subjects of passivised transitives and unaccusative intransitives would have their lower subject dependency labelled by VP-V'-DP-right (because they are generated as complements to V rather than specifiers to v).

Note that because in the Xbar tree adjuncts are defined as having identical parent and head-child categories, adjunction and complementation are automatically distinguished without the need to separately identify this distinction with a complement feature (-C in Collins (1999)). Coordination is treated as complementation in MGbank, meaning that it too is distinguished from adjunction here, although it is not distinguished from other types of complementation. Finally, as was also noted in section 5.4.4.1, where the head word of a given non-terminal is null, the system recursively searches into the null item's complement until it finds an overt word to use as the head. Thus all dependencies are defined between overt words only. In the case of the unlabelled dependencies, this can lead to duplicate dependencies (i.e. more than one dependency between the same two overt words), and these are removed.

The unlabelled, undirected dependencies simply encode that there is some dependency between word A and word B, without any indication of the structural configuration in which this dependency was found or even which word is the head and which the dependent; this type of dependency is therefore clearly the easiest to recover. Of intermediate difficulty between labelled directed and unlabelled undirected are the directed, unlabelled dependencies which again omit the structural configuration of the dependency by omitting the label, but do indicate which word is the head and which the dependent.

The first point to note about the results in table 5.2 is that the exact match scores are very low, just 17% for labelled directed dependencies (the exact match score for the Xbar phrase structure trees themselves was 16%). This means that the majority of trees in MGbank have at least one error in them. The exact match scores for the semantic dependencies are a little higher in the case of both unlabelled dependency types (33% vs 24%), which is understandable given that in certain respects the semantic dependencies are more forgiving than the syntactic ones. For example, if a temporal adjunct attaches to vP instead of to TP, and if T is occupied by an overt auxiliary, then

the syntactic dependency score will be penalised whereas the semantic one will not. This is because the dependency extracted will be between the head of the adjunct and the lexical verb in the main structure, rather than between the head of the adjunct and the auxiliary verb as it would have been on the syntactic side assuming the correct attachment to TP. But because in semantic terms it is the lexical verb which is the head of the main clause anyway, the dependency will be between the head of the adjunct and the main lexical verb irrespective of the presence of an overt auxiliary and of whether attachment is to TP or vP, meaning that this attachment error will not be penalised here.

The F1 scores give a much better indication of the overall quality of the structures in MGbank. Labelled syntactic dependency recovery is at 84.3% vs 80.07% for semantic dependencies. Here then, the situation is reversed, with syntactic dependencies apparently more forgiving at this more granular phrase-level of analysis. The reason for this probably has to do with the fact that the semantic dependencies are often less local than the syntactic ones. For example, a discourse PP attached to CP headed by overt *that* is syntactically dependent on the word heading the CP it is adjoined to but semantically dependent on the lexical verb embedded several phrases down. This means that as long as the PP is attached to the CP headed by *that* it will not be penalised on the syntactic side, whereas this is not the case on the semantic side: suppose, for instance, that for some reason *that* fails to c-command the correct lexical verb. Then the PP will also fail to be semantically dependent on that lexical verb resulting in penalisation on the semantic side but not the syntactic side.

The unlabelled, undirected dependencies yield the highest score for obvious reasons: 94.73% for the syntactic dependencies and 93.15% for semantic dependencies, which is high given that these scores include many types of movement dependencies. From the perspective of the correct recovery of predicate argument structure, however, it is the directed unlabelled dependencies which are the most important of the tree dependency types. This is because it is often crucial to know which item is the predicate and which the argument or which the modifier and which the modified, but the syntactic labels are not required for semantic interpretation. On the syntactic side, these directed unlabelled scores are only slightly lower than their corresponding undirected scores at 93.58%. Of course, if the recovery of predicate-argument structure is the goal, then the semantic dependencies are the most relevant, and fortunately, at 90.87%, these are also much closer to their unlabelled, undirected counterpart scores than to the labelled directed ones.

### 5.6.2 Error analysis of global dependency recovery

A manual inspection of the trees themselves reveals that the majority of the errors are relatively minor and would not impede subsequent inference of the correct predicate argument structure. For example, in the sentence *Otherwise actual profit is compared with the num-day estimate*, the discourse-related adverb ‘otherwise’ was attached to TP by the parser rather than to CP as is the case in the gold structure. For all hand-crafted trees in MGbank, discourse adverbs were adjoined to CP wherever possible (in line with Minimalist theory), but sometimes this was not possible owing to word order considerations. For example, in the sentence, *The gene thus can prevent a plant from fertilizing itself*, the subject precedes the discourse adverb ‘thus’. In MGbank such word orderings are generated by assuming that the subject undergoes topicalization to spec-CP, but because adjunction is necessarily outside of specification, this means that the discourse adverb must be adjoined to some phrase inside of the CP to which the subject moves, and in MGbank the phrase to which it adjoins in such scenarios is TP. This means that during the automatic annotation phase both options are available to the parser and there is nothing in either the PTB or CCGbank which would cause the system to prefer one over the other.

Additional heuristic rules could be used to fix these errors during automatic tree-banking, of course. However, nothing much rests on this attachment difference from the point of view of semantic interpretation, and indeed the semantic unlabelled dependencies (and even the syntactic unlabelled ones in the absence of an overt complementizer or auxiliary verb) would not be affected by this error, because here the dependency between the adverb and the main lexical verb would still be recovered. Similar considerations apply to the sentence *The company said it made the purchase in order to locally produce hydraulically operated shovels*, where the automatic generator attached *in order* to the TP of the purposive *to* clause, rather than to the CP dominating it, as is the case in the gold test set (some evidence that *in order* should be attached to CP and not TP comes from infinitivals with the complementizer *for*, in which case *in order* precedes this complementizer, as in *staff must be committed to the change in order for it to succeed*).

Another common error made by the automatic generator which is similar to that noted above was that it attached a temporal adjunct such as a PP or adverb to vP rather than to TP. An example of this was that in the sentence *Ten points of the drop occurred during the last num minutes of trading*, the temporal PP *during the last num minutes*

*of trading* was attached to the vP whose extended projection is headed by ‘occurred’ rather than to the TP dominating it. Again, adjunction of temporal adjuncts to both vP and TP had to be permitted in the seed corpus owing to various word orders which were encountered, although wherever possible TP was preferred to make the trees closer in spirit to Minimalist assumptions. The use of null heads makes reconstructing the perfect trees challenging, and these sorts of very common errors are a primary reason for the very low exact match scores.

Another example of an error which is more serious was the treatment of a passive participle as an adjective, with the passive auxiliary *be* also being incorrectly treated as the copula. This occurred in the phrase *Management’s total could be reduced*, for instance. In this seed set that the supertagger was trained on, the word form *reduced* occurs just once, as an attributive adjective in the phrase *at reduced prices*. This is likely the root cause of this error, although a quick search for the string *be reduced* in the automatically generated portion of the corpus reveals 5 further instances of this substring all of which are correctly analysed as passives, so this error is curious.

The supertagger also seems to disprefer analyses involving a purposive *to* adjunct clause, instead opting to tag the main verb as a raising or control verb. This was the case for the verb *using* in the sentence *They renamed it Swiss Cantobank and are using it to expand abroad*, for instance, where *using* was tagged as an object control verb, while in the sentence *jumped 527.39 points to close at 34996.08 jumped* was analysed as a raising verb (with *527.39 points* an adjunct). These types of errors arise because the C&C supertagger used for the treebanking was trained on the small and somewhat skewed dataset of the 800 or so hand-crafted seed sentences. In some cases an MG category was used only a handful of times or even once in this seed set, making it very difficult for the supertagger to predict it accurately at test time. This was also the case for light verb categories, for instance. For example, the gold test set analyses *get* in *get a chance* (*wsj\_2300.mrg*, line 49) as a light verb, placing it in the little v position, with the DP *a chance* occupying the position that the VP normally would. In the automatic tree, however, *get* was tagged as a standard transitive verb and therefore appears in the V position.

The supertagger also made a number of other errors, some of which had knock on effects for other parts of the structures containing them. For example, in the sentence *It continues to gain strength in the chamber but remains far short of the two - thirds majority required to prevail over Mr. Bush*, the coordination is carried out at the vP level in the gold structure, but the supertagger clearly assigned a much higher score

to the TP coordinator supertag for *but*, meaning that that the TPs were coordinated instead. In order to allow for the subject *it* to be shared by both conjuncts, the parser was then forced to topicalise the subject out of spec-TP of both conjuncts (using ATB movement) to spec-CP, and this unnecessary movement therefore created additional structure in the CP that was not contained in the gold tree. Both the movement and the additional structure would clearly have impacted the scores of the labelled dependencies, and even the syntactic unlabelled ones because in its moved position the subject would become a syntactic dependent of the coordinator. The semantic unlabelled score would not be affected owing to the fact that *it* would be correctly analysed as a dependent of the two verbs *continues* and *remains* in both its spec-TP and spec-CP position (because conjuncts and not the coordinator are the *semantic* heads of the coordination phrase).

The correct predicate argument structure in this example could thus clearly still be derived in spite of the error, albeit with an unwanted element of (topicalization) discourse meaning also encoded by the structure. A more serious issue in this sentence was the fact that the supertagger also tagged *far* as a preposition rather than an adjective, and *remains* as a verb taking a PP complement, with the PP *of the two - thirds majority required to prevail over Mr. Bush* being adjoined to the erroneous PP headed by *far* rather than it being a complement of *short* which it incorrectly analysed as a noun. It is not entirely clear why the supertagger made each of these errors. It is probable that in reality it made one genuine error and that this had a knock on effect by precluding the correct tags for several other words being assigned. What this confirms is that an improved supertagger could lead to significant improvements in the quality of MGbank.

At least one error arose owing to an error made in the original annotation of the seed set on which the supertagger was trained. Specifically, the verb *stop* in the sentence *there 's some price at which we'd stop bidding* (*wsj\_745.mrg*, line 10) had been annotated as being an unaccusative verb (i.e. with the feature INTRANS) (as it arguably should be in a sentence like *the river level will stop falling*). Although this has now been corrected in the seed set, the treebank generator would need to be rerun to remove its unwanted effects from the automatically generated portion of the corpus. One of these effects seems to be evident in the sentence *Initially the company said it will close its commercial real-estate lending division and stop originating new leases at its commercial lease subsidiary*, where *stop* was incorrectly annotated as INTRANS, meaning that its subject was base-generated inside VP rather than vP, leading to an er-

ror in the labelled directed dependency scores on both the syntactic and semantic sides. Again, however, this error is not so serious as to preclude a reasonable semantics being derived (because in either case the subject argument is base generated in a structurally superior position to the clausal/gerund argument *originating new leases*).

One class of errors arose owing to the fact that the treebank is currently undergoing revision. Specifically, copula constructions were originally analysed as involving a direct adjP, PP or DP complement to the copula verb, whereas MGBank is currently undergoing a transition to an improved analysis where a null [predicator] head first takes the adjP, PP or DP as its complement, before the copula selects the resulting prdP as its complement. This makes coordination of unlike constituents following copula verbs (*he is happy and in the garden*) straightforward, and also allows the subject to be base generated in spec-prdP, enabling the generation of structures in which the thematic subject remains in-situ, with an expletive occupying spec-TP (*there are several men in the garden*). The gold test set was annotated using the new analysis, but the old analysis was the only one implemented when the treebank was being generated, hence it is the only one present in the automatically generated portion of the corpus, and this led to an error in the sentence *Third-quarter sales of U.S. credit services were disappointingly below sales of a year earlier Dun & Bradstreet said*.

There were also a number of NP attachment errors. For example, for the phrase *organization men in grey - flannel suits*, the system failed to analyse *organization men* as a constituent; instead, it predicted that *men in grey - flannel suits* was a constituent, with *organization* left adjoined to it. Again, this would not seriously compromise semantic interpretation, although other adjunction/coordination attachment errors were more serious. For instance, in the sentence *K mart officials and Mr. Pilevsky would n't comment on the sale*, the system failed to analyse *K mart officials* as a constituent, instead treating *officials* as a conjunct and *mart* and *k* as separate adjuncts to the coordinate phrase. Given that the parser's search space was constrained so as to prefer constituents which were found in either the PTB or CCGbank, this type of error may seem surprising. However, many exceptions to these constraints had to be allowed in order to avoid blocking various other structures, and these exceptions sometimes allow errors such as this to find their way into the treebank.

In other cases these types of errors arose because of mistakes in CCGbank and/or the PTB. For example, in the phrase *housing and urban development secretary Jack Kemp*, *Jack Kemp* was not analysed as a constituent. Instead, *Jack* was analysed as a rightward adjunct to the phrase *housing and urban development secretary* and *Kemp*

was analysed as a rightward adjunct to the larger phrase *housing and urban development secretary Jack*. In the PTB, *Jack* and *Kemp* are sisters, but they have as a third sister the phrase *housing and urban development secretary*, meaning that they do not constitute a constituent. In CCGbank, *Jack* and *Kemp* do form a constituent, but the entire string *housing and urban development secretary Jack Kemp* is erroneously analysed as exclusively right branching meaning that the substring *housing and urban development secretary* is not analysed as a constituent (instead *housing* is analysed as the first conjunct and *urban development secretary Jack Kemp* as the second conjunct).

The complement/adjunct distinction for PPs is notoriously murky and difficult for parsers and this inevitably shows up as a further type of error in the output of the automatic generator. For example, in the phrase *the charter in its current form*, *in its current form* was analysed as a complement of *charter* (meaning that *charter* was erroneously tagged with the category for a noun taking a PP complement), rather than as an adjunct to the NP *charter* (or to the DP *the charter*). This kind of error is understandable given that the PTB makes no distinction between PP complements of N and PP adjuncts of NP, treating both in structural terms as adjuncts (i.e. as both daughter and sister to an NP node).

Another class of errors occurs because, for efficiency reasons, certain null heads were held back from the chart unless a parse could not be found, as described in section 4.7.4. One such class of null heads are the [extraposer] heads which map various types of phrase into rightward moving versions of themselves. For the phrase *sharper losses than Tokyo*, the automatically generated tree analysed *than Tokyo* as a complement of *losses*, whereas in fact it is clearly some sort of dependent of *sharper*. This error is understandable as this is a difficult dependency to recover, and in fact the PTB incorrectly analyses *than Tokyo* as an adjunct dependent of the NP headed by *losses*. In the MGbank gold test set, the underlying structure for this phrase is *[[sharper than Tokyo] losses]*, with *than Tokyo* initially a complement of *sharper* before undergoing rightward movement to TP to derive the final surface word order (one piece of evidence for this structure is that it is possible to place a temporal adjunct between *losses* and *than Tokyo*, e.g. in *Hong Kong experienced sharper losses this morning than Tokyo*). But because there was a parse available before the [extraposer] head was ever introduced, the correct structure was never able to be derived.

A final set of errors arose because MGbank does not yet have complete coverage and it was necessary to add a few categories to the inventory while annotating the gold test set by hand. Because these categories were simply not available to the supertagger

when the trees were being generated automatically, there was no way that the correct structures could have been derived. For example, for the phrase *35 cents to 40 cents a share* (wsj\_2396.mrg, line 2), a new ditransitive preposition category was introduced for *to* allowing this word to take *35 cents* as a specifier and *40 cents* as a complement. Another example was that the sentence *Housing and Urban Development Secretary Jack Kemp called on the Federal Reserve System to lower interest rates* required a new type of particle object control verb for *called*. There were five of these types of errors made by the system.

### 5.6.3 Targeted evaluation of some long-distance dependency constructions in MGbank

The previous section provided an evaluation and error analysis for overall dependency recovery in MGbank. However, overall dependency scores are inadequate for assessing the performance of parsers on specific types of constructions, particularly those which are found in the long Zipfian tail of rare construction types (Rimell et al., 2009). This is because there are usually too few examples of any single given construction to have any noticeable impact on the overall dependency scores. Many of these constructions involve various types of (bounded and unbounded) long-distance dependencies, and it is these which motivate the use of parsers based on grammatical formalisms which go beyond context free levels of expressivity. This section therefore provides a targeted evaluation of MGbank's quality with respect to a selection of these constructions, beginning with various A-movement constructions (specifically, passivisation and various infinitival constructions), before moving onto consider a selection of A'-movement constructions (specifically, various types of relative clause).

In order to assess the parser's precision and recall on individual construction types, it was necessary to manually inspect its output. This process is clearly very time-consuming and for this reason, the evaluation is not exhaustive, but rather focuses on a selection of linguistically interesting constructions. The idea of this evaluation is to give the reader a rough idea of the current state of MGbank at present. To keep the evaluation manageable, the test set for each construction was limited to including all sentences in the automatically generated portion of MGbank up to and including the 20th sentence containing the construction in question. Except for relative clause constructions, each test set starts with the first sentence of section 00. However, note that because many of the relative clauses in section 00 had been annotated manually



already, the test sets for the relative clause constructions begin at the start of section 01 of MGbank rather than at the start of section 00.

Constructions were identified either using simple regular expression matching over the PTB tree, or, in cases where this was not possible, a search was conducted over the PTB tree to identify the specific configuration in question. Recall for each test set of 20 sentences was then computed in the standard manner, i.e. as: true positives / (true positives + false negatives). A true positive here describes a situation where the parser both identifies that a given construction is present by identifying the primary category used in that construction (e.g. it correctly predicts a subject control verb), and correctly analyses all the key aspects of the surrounding structure (e.g. it must also predict that the null controlled subject is an A-movement trace, not NOC [pro-d]). Conversely, a false negative could mean that the parser failed to predict the main category entirely, or that it got some of the surrounding structure wrong.

In order to compute precision, it is necessary to know in how many instances the parser incorrectly predicted a given construction to be present, or correctly predicted it (by correctly predicting the main category used in that construction) but misanalysed some part of its structure. This was achieved by using Autobank's search facility to search for lexical categories which appear in the construction in question. If the category that best identifies that construction was a verb, then all morphological forms of that verb (i.e. 3rd singular present, non-3rd singular present, progressive, past/perfect participle, passive participle, bare) were searched for. Precision was then computed as: true positives / true positives + false positives. A false positive describes a situation where the parser predicts the main category for the construction being evaluated but that construction is not actually present. For example, if the parser predicts a subject control verb when it should have predicted a subject raising verb, then this would count as a false positive for the evaluation of subject control.

Note that the error analysis was carried out exclusively with reference to the Xbar trees, rather than to the MG derivation trees which contain the additional fine-grained subcategorization information which the Xbar (and MG derived) trees lack. This makes the analysis somewhat more forgiving than would otherwise be the case. For instance, it does not matter whether a PP is identified as temporal, locative or agentive (i.e. whether the P head bears the feature TMP, LOC or AGENT), as long as it is adjoined to, or is taken as a complement by, the correct phrase/head.

### 5.6.3.1 Passivization

Recall: 18/20

Precision: 18/18

To construct the test set of 20 sentences, passive verbs were first identified by searching for all those MGbank sentences whose PTB bracketings contained the past participle tag VBN followed by an empty category (-NONE-); any incorrect matches were then filtered out by manual inspection and the first 20 sentences in MGbank which contained genuine passives were retained. To be considered a genuine passive, the PTB bracketing had to include the object trace of the passivisation. For example, in the following example from the test set

- (340) The exact amount of the refund will be determined next year based on actual collections made until Dec. num of this year.

the verbs *determined* and *made* are analysed as past participles with extracted objects in the PTB, making them genuine passives for current purposes, while the verb *based*, although analysed as a past participle in the PTB, lacks any object trace there and so is not counted as a genuine passive here.

The 20 examples of passivisation in the test set included simple passivised transitives (*Legislation to lift the debt ceiling is ensnared in the fight over cutting capital-gains taxes*), ditransitives (*Shorter maturities are considered a sign of rising rates*), prepositional datives (*food and drinks are banned to everyday visitors*), passivised ECM verbs (*the senate isn't expected to act until next week*), and reduced passives lacking the auxiliary, both in relative clause contexts (*The plant will produce control devices used in motor vehicles*) and comparative contexts (*The total was far higher than expected*) (all of these examples were correctly analysed as passives in the automatically generated trees). To be correct, the Xbar structure had to include passivisation from the correct base generated position to the correct surface position, and the passive auxiliary, where present, had to be correctly identified as such. Furthermore, if an agentive *by* phrase was present, it had to be adjoined to vP.

The two cases where the parser had failed to identify a passive both involved reduced relative clauses where the supertagger had mistakenly assigned the category of an adjective to the past participle and then adjoined that adjective as a post-nominal modifier to the DP (as occurs legitimately in examples like *Pierre Vinken, num years old, will join the board as a non-executive director*). An example of where this oc-

curred here was with respect to the verb *sold* in the following sentence (the PP *at Monday's auction* was also adjoined to the DP).

- (341) The yield on six-month Treasury bills sold at Monday's auction, for example, rose to 8.04% from 7.90%.

On the other hand, for the following example

- (342) The plant will produce control devices used in motor vehicles and household appliances.

the automatically generated tree correctly analysed *used* as a passive in a reduced relative clause, so it is not the case that these were beyond the reach of the parser. However, because the [relativizer] null heads which are required to generate such structures are held back from the chart unless another parse cannot be found, the supertagger must assign a low enough score to the adjective tag for the past participle verb so as to prevent it from entering the chart and enabling an incorrect analysis which bleeds the correct one.

### 5.6.3.2 *Want*-type subject control

Recall: 17/20

Precision: 17/24

The MGbank analysis of subject control verbs like *want* in *Jack wants to help Mary* was discussed in section 3.3.6. Note that although control into gerund complements (*Jack tried eating some cake*) is also part of the MGbank grammar, only infinitivals are considered here.

To be judged as correct, the control verb had to take a *to*-infinitival CP as a complement, and the subject had to be extracted from the correct argument position within this clause first via spec-vP of the clause containing the control verb (where it picks up its second theta role) and then to the surface subject position (i.e. spec-TP). In cases where the complement of the control verb was a coordinated CP, then the subject had to be extracted in ATB fashion from both conjuncts. If the null controlled subject was misanalysed as an NOC [pro-d] head, rather than as a trace of A-movement, then the construction was judged as incorrect. An example of where the parser correctly performed such ATB extraction was the following sentence, where *it* was correctly extracted from both the *obtain* and *complete* conjunct clauses.

- (343) The thrift holding company said it expects to obtain regulatory approval and complete the transaction by year-end.

Of the three instances of control in the test set which the parser failed to correctly analyse, two were analysed as raising verbs, with the subject correctly extracted from the (TP) clausal complement. The error was therefore down to the supertagger incorrectly assigning a higher probability to the raising verb tag than to the correct subject control one. In one of these instances (*wsj\_0006.mrg*, sentence 26), the verb was *beginning*, which arguably should be analysed as a raising verb, as it can appear with an expletive subject (*there were beginning to be rumours of an imminent coup*). In general, however, the PTB does a good job of distinguishing control from raising, which is not too difficult given that raising verbs are drawn from a fairly closed class (unlike control verbs, which are an open class).

In one example (*wsj\_0034.mrg*, sentence 13), the parser incorrectly analysed the controlled subject as the NOC [pro-d]. This happened because a [focalizer] head was needed for this sentence, and so the parser, being unable to find a parse initially, began to introduce the additional null heads into the chart (see section 4.7.4), one of which was [pro-d]. However, as noted in section 5.4.4.2, one of the heuristics used to prune candidate trees was that those with more null pronominals should be dispreferred (and this rule is applied before the one which prefers less movement) and so this tree should have been pruned. However, after as many trees as possible had been generated using the atomic MG categories, another supertagger was trained using the complex MG category approach to be described in the next chapter. In this approach, all null head are anchored to overt MG heads inside LTAG-like MG categories. This means that the method for holding back null heads from the chart no longer applies, and the category for a verb which has a null [pro-d] anchored to it is different from that which does not, meaning that it is possible that the former was assigned a higher probability than the latter in this instance with the result that only it made it into the chart.

All 7 of the instances where the parser incorrectly predicted subject control were in fact instances of raising. 5 of these involved the verb *have*, an examples of which is shown below.

- (344) It has to be considered as an additional risk for the investor said Gary P. Smaby of Smaby Group Inc. Minneapolis.

The reason for this is that the hand-annotated portion of the corpus contains multiple examples of where this verb was incorrectly analysed as a control verb. Correcting

these annotations regenerating the automatic portion of MGbank would resolve this issue for these cases.

### 5.6.3.3 Subject raising verbs

Recall: 12/20

Precision: 12/14

MGbank's analysis of raising verbs was also discussed in section 3.3.6. To be judged correct, the complement clause of a raising verb like *seems* in *Jack seems to like Mary* must be a bare TP rather than a CP as was the case with subject control, and the controlled subject must be extracted from the correct base position within that TP. Furthermore, the extraction must proceed immediately to the surface subject position, bypassing the intermediate spec-vP (theta) position which was targeted in the case of control. The reason for this is that while *Jack* is a wanter in *Jack wants to help Mary*, he is not a seemer in *Jack seems to help Mary*.

All 8 examples missed by the parser were analysed as involving subject control rather than raising (again, owing to the supertagger assigning a higher probability to the incorrect tag), and 5 of these involved the raising verb *have*, an example of which is shown below.

(345) The beds at the Bowery Mission seem far drearier when he has to tuck a little girl into one of them at night.

As noted in the previous section, this is because this verb was misanalysed as a control verb during the manual annotation of MGbank and this error has been replicated by the supertagger.

Regarding the two incorrect predictions of subject raising made by the parser, one was in fact an instance of obligatory subject control, while the other was an instance of adjunct control; this latter example is shown below.

(346) The filing on the details of the spinoff caused Cray Research stock to jump \$2.875 yesterday to close at \$38 num in New York Stock Exchange composite trading.

### 5.6.3.4 Exceptional Case Marking (raising to object)

Recall: 15/20

Precision: 15/21

Section 4.7.3 discussed the MGbank raising-to-object analysis of ECM verbs. In order for the construction to be judged correct, the ECM object must raise to spec-VP of the ECM verb (to check accusative case) from the correct embedded argument position. Unlike in the case of object control, this DP does not stop off first in an inner spec-VP position (hence it receives no theta role from the ECM verb). The infinitival complement clause of the ECM verb must also be a TP, rather than a CP as is the case for object control. ECM verbs are differentiated from object control verbs in the PTB in that in the former case the ECM object remains inside the embedded S clause as its subject, whereas in the case of object control the object is extracted to the matrix VP, leaving behind a trace in the embedded subject position. However, passivised ECMs are not differentiated from passivised control verbs in the PTB (unlike in the MGbank grammar) and hence were excluded from the evaluation.

All 5 cases of ECM examples missed by the parser were mistakenly analysed as involving object control. Conversely, of the 6 examples incorrectly predicted to be ECM by the parser, 5 were in fact object control, while 2 were actually purposive *to* clauses involving adjunct control, an example of which is shown below.

(347) It said it has taken measures to continue shipments during the work stoppage.<sup>8</sup>

Again, all of these errors are clearly due to the supertagger failing to assign the highest probability to the correct ECM category for the verb in question. It is worth noting, however, that whereas subject raising verbs are usually easy to distinguish from subject control verbs and hence are annotated very reliably in the PTB, in the case of ECM (raising to object) vs object control the distinction is not nearly so clear cut because ECM verbs unlike subject raising verbs are not a closed class, and it is not always clear whether the object is a PATIENT argument of the embedding verb or not. As a consequence, there are many cases where a verb could be analysed as either one of these verb types, hence the distinction becomes essentially arbitrary.

The basic distinction between ECM and object control is that only an object control verb assigns a theta role to its DP object. Thus, in the ECM example *Jack expected Mary to help Tim*, it is the whole proposition *Mary to help Tim* which is expected, and *expect* assigns no separate theta role to *Mary*, hence one can transform this example into the pseudo-cleft *what Jack expected was Mary to help Tim*. In the superficially similar object control example *Jack persuaded Mary to help Tim*, on the other hand,

<sup>8</sup>The *to* clause in this sentence is analysed in the PTB as an adjunct clause, although arguably a better analysis would be that *take measures* is a light verb construction that is equivalent to a subject control verb.

it is *Mary* who is being persuaded, not the proposition *Mary to help Tim*, which is the thing *Mary* is persuaded of. Hence *persuaded*, unlike *expect*, must assign theta roles to both the DP and the CP, and as a consequence a pseudo-cleft is not possible here: *\*what Jack persuaded was Mary to help Tim*. Another way in which the difference shows up is that ECM verbs can take expletive *there* as their object (because expletives do not bare theta roles), whereas object control verbs cannot (*Jack expected there to be trouble* vs *\*Jack persuaded there to be trouble*).

Clearly, only a person can be persuaded, not a proposition, hence in this case the object control analysis is forced. However, there are other cases where it is not clear whether the verb is assigning an additional theta role to the DP object or not. One example from the ECM test set was the following sentence.

(348) Because of budget constraints in Washington the U.S. encourages Japan to share economic burdens in the region

For this example, the PTB analysed *encourages* as an ECM verb. It is certainly the case that, while one cannot really talk about persuading a situation, it seems much more acceptable to talk about encouraging a situation, and as a consequence the verb *encourage* does seem more amenable to appearing with a *there* object than *persuade* does.

(349) Jack expected/??encouraged/\*persuaded there to be discussion on the matter.

As indicated, however, such examples still seem far less acceptable than those with an unequivocally ECM verb like *expect*. The same situation holds for pseudo-clefting, as illustrated below.

(350) what Jack expected/??encouraged/\*persuaded was Mary to help Tim.

This verb therefore seems to be a marginal case, but of course the annotators of the PTB were forced to make a choice, and they opted to label it as an ECM verb. An example from the test is shown below.

(351) Because of budget constraints in Washington the U.S. encourages Japan to share economic burdens in the region.

The parser ‘incorrectly’ analysed this sentence involving object control. However, given the above considerations, this is arguably not really an error at all, despite being recorded as one in the scores. Interestingly, *encourages* occurs only once in the seed

set of hand-annotated sentences on which the supertagger was trained, and is annotated there as an object control verb (the other ECM test cases which the parser misanalysed were less ambiguous than this one, involving verbs like *authorise* and *enable*, but neither of these verbs occurs in the seed corpus at all). The question which arises, therefore, is why the supertagger made this error.

Recall from 5.4.1 that during preprocessing, ECM verbs were marked with AGRS instead of ARG1, in order to distinguish them from object control verbs. This was achieved by marking any verb whose clausal argument was ARG1 with the AGRS tag. This was intended to distinguish ECM from object control verbs, whose clausal arguments are generally ARG2 (with their DP objects being ARG1). Unfortunately, it became clear as this analysis was being conducted that the Propbank annotators were not consistent in this regard. For instance, in the following sentence

(352) Judge Keenan also directed the prosecutors to show that Mrs. Marcos 's Fifth Amendment right against self-incrimination won't be violated.

*the prosecutors* is marked as ARG1 and the infinitival clause as ARG2, whereas in following sentence

(353) It also asks them to add two-sevenths and three-sevenths.

*them* is marked as ARG2 and the infinitival clause is marked as ARG1. This fact clearly resulted in the supertagger confusing ECM and object control to a considerable extent and in future some other way must be found to distinguish them (only object control involves movement of the DP out of the lower clause in the PTB, so it would quite straightforward to use the presence of a trace in the embedded subject position to achieve this).

### 5.6.3.5 Object Control

Recall: 9/20

Precision: 9/45

The MGbank analysis of object control was discussed in section 4.7.3. To be judged correct, the null controlled subject had to be extracted from the correct position inside an infinitival CP (rather than a TP, as is the case for ECM) embedded clause and move first to the inner spec-VP of the control verb (picking up its second theta role) and then to its outer spec-VP (to check accusative case).



Of the 11 instances of object control which the parser missed, all 11 were misanalysed as ECM. In one of these cases, the controlled null argument was also misanalysed as NOC [pro-d]. The precision score is very low here here, and again, in most cases this was due to a sentence which was in fact an ECM being predicted to be a case of object control. The difficulty with distinguishing between ECM and object control was discussed in the previous section. In 5 of the 13 incorrect predictions, however, the infinitival clause was in fact a purposive. An example of this is shown below.

(354) Analysts expect Armstrong to use proceeds of the sale to reduce debt buy back stock or perhaps finance an acquisition.

The verb *use* does not appear before infinitival *to* in any of the examples in the hand-annotated training data, hence the supertagger had to make a guess based on the surrounding context and did so incorrectly.

Although the precision and recall scores for object control are very low, it is perhaps worth noting that the distinction between these two constructions is somewhat theory internal (CCG, for example, does not distinguish between them in the syntax, only in the semantics) and in most cases the arguably semantically most crucial aspects of both the ECM and object control examples (i.e. the extraction and landing site of the DP) were correctly recovered. If these two constructions were treated as one for the purposes of this evaluation, then the recall on the 40 test sentences would be a much more impressive 38/40.

### 5.6.3.6 Control into infinitival purposive clauses (adjunct control)

Recall: 15/20

Precision: 15/22

As was discussed in section 3.3.7.2, control into adjuncts in MGbank involves ATB phrasal movement and the unification of the mover inside the adjunct with its identical copy inside the main structure at the point at which the main and adjunct clauses are merged. Infinitival *to* clauses can serve as purposive adjunct clauses. One example from the test set is shown below.

(355) Mr. McAlpine resigned to pursue a consulting career.

In this example, *Mr McAlpine* obligatorily controls the null subject of the infinitival adjunct clause (that the infinitival clause is an adjunct, rather than a complement of

the verb *pursue*, is confirmed by the fact that removing this adjunct changes neither the acceptability nor the meaning of the main clause). In order for such examples to be judged as correct for this evaluation, the correct extraction site in both the main clause and the adjunct clause had to be identified, i.e. the subject in spec-TP had to be correctly co-indexed with its two lower traces, one in spec-vP of *resign* and one in spec-vP of *pursue* in example 355) and the infinitival clause had to be an adjunct to vP. Given the adjunction configuration, the only way the aforementioned co-indexation configuration could be achieved is by ATB phrasal movement.

Of the 5 examples in the test set which the parser missed, 4 were analysed as involving NOC PRO inside the adjunct clause rather than a trace of ATB movement (see section 5.6.3.2 for discussion of how this might have occurred). Note that a PRO analysis of obligatory control, including adjunct control, is in fact the standard analysis in Minimalism (though the movement theory of control is adopted by a substantial minority of Minimalists), and such an analysis by the parser would not preclude the correct semantics being generated (though additional semantic interpretation rules would be required to co-index PRO with its obligatory antecedent). These errors can therefore be considered as relatively minor.

A more interesting error was found in the following example, in which the verb *move* is used as a subject control verb having two infinitival clause dependents: a complement infinitival and a topicalised adjunct infinitival.

(356) To capture the investment Southeast Asian nations will move to accommodate Japanese business.

The parser incorrectly topicalised the leftmost infinitival from the complement position, rather than from the correct adjunct position, and analysed the actual complement clause as being a purposive. The supertagger is not to blame here as it assigned the correct subject control category to *move*. Instead, it is the parser itself which was at fault as failed to insert the two infinitivals into their correct base positions. Of course, a full statistical model of the derivation itself could help here, as this would be able to learn that topicalisation of adjunct infinitival clauses is much more likely than topicalisation of complement infinitivals (though such topicalisation does seem to be possible: *Jack said he would try to help Mary, and to help Mary he has indeed tried.*). However, there simply is not enough seed data to train such a model. One could potentially use the PRP or PNC tags which are found on infinitival purposive clauses in the PTB, although this would not be a simple matter of including them on the CCGbank tag for

the verbs or *to* particles in these clauses as these have the same (atomic) MG category regardless of whether the infinitival is a complement or an adjunct.

6 of the 7 examples incorrectly analysed as purposives were in fact infinitival relative clauses. In defence of the parser here, at least two of these sentences were genuinely ambiguous between a purposive and relative reading. One of these is 357 below.

(357) Texas Instruments Japan Ltd., a unit of Texas Instruments Inc., said it opened a plant in South Korea to manufacture control devices.

The PTB analyses this sentence as involving rightward movement of the infinitival relative clause from inside the relativized NP *a plant*, with *a plant* therefore being the subject of *manufacture*. To my ears, however, this infinitival clause is just as naturally interpreted as a purposive modifying *opened*, with *it* as the subject of *manufacture*.

The final example where the parser incorrectly predicted a purposive involved a *for-to* clause which had been extraposed away from an expletive *it* subject (such constructions are handled using rightward movement in MGbank). The sentence in question is shown below.

(358) Judge Ramirez num said it is unjust for judges to make what they do.

### 5.6.3.7 *for-to* infinitivals

Recall: 11/20

Precision: 11/11

As was discussed in section 4.7.3, the MGbank analysis of infinitival clauses headed by the prepositional complementizer *for* involves a C-P shell structure in which *for* is initially generated as P head that takes the infinitival TP as its complement and attracts the subject to its specifier (checking the latter's accusative case feature), before a null C head takes the resulting PP as its complement and triggers P-to-C head movement of *for* (thus accounting for the dual prepositional/complementizer nature of this item), which correctly places *for* to the left of the subject. The resulting CP is then selected as a complement by the relevant embedding verb (*he asked for her to help*), adjective (*the stage was set for her to begin*), or, in the case of *for-to* relatives (*a pen for her to write with*), by the higher [rel] C head. In order for this construction to be judged as correct for this evaluation, all of these aspects of the construction had to be correct and the infinitival subject had to be extracted from the correct base-generated position.

While the precision on this construction is 100%, the recall is contrastively very low at just over 50%. This means that whenever the parser predicts this construction it does so correctly, but it misses about half of the instances of it in the PTB trees covered by MGbank. All of the 11 missed examples involved *for* being misanalysed as a standard preposition, hence these were ultimately supertagging errors. For example, consider 359 below.

(359) What becomes custom in the Bush administration will only become more difficult for future presidents including Democrats to undo.

In this example, *for* was analysed as a standard preposition governing the infinitival relative clause, rather than as the complementizer within that clause. Furthermore, the two arguments of that infinitival clause were generated in the wrong positions, with *what becomes custom in the Bush administration* base generated in spec-vP of the *undo* clause (making it the undoer rather than the thing undone) and *future presidents* was base-generated as the object complement of *undo* (making it the thing undone) before being relativized. In the correct PTB analysis, it is *what becomes custom in the Bush administration* which is the relativized argument that subsequently undergoes further movement to its surface subject position.

A further 4 cases involved the infinitival being treated as a purposive clause. One of these sentences is shown below.

(360) It would be sad for Mr. Gonzalez to abandon them to appease his foes.

The parser analysed this example as involving *for Mr. Gonzalez* as a regular PP taken as a complement by the adjective *sad*, with *to abandon them to appease his foes* a purposive clause modifying *be*. Again, the supertagger is ultimately responsible for these errors and it is likely that the root cause of this is the fact that there are simply many more examples in the small training set on which it was trained where *for* occurs as a standard preposition.

### 5.6.3.8 Infinitival relative clauses (subject and object)

Recall: 7/20

Precision: 7/14

The promotion-based MGbank analysis of restrictive relative clauses was discussed in section 4.2.1. The examples used in that discussion involved finite relative clauses

but the analysis of infinitival relatives such as *the next country to be removed* with a relativized argument, or *the best way to do it* with a relativized adverbial NP, are generated in the same manner. The test set used here included 15 cases where the relativized item was the subject of the relative clause and 5 cases where it was an adverbial NP of that clause; there were no cases of relativized objects present in the test set, these being much rarer than subject and adverbial infinitival relatives. In order to be considered correct, the relativized item had to be extracted from the correct argument or adjunct position, and appear in the correct surface position in the left periphery of the relative clause. None of the examples in the test set had an overt *wh* item present (for infinitival relative clauses, overt *wh* items are only permitted when a preposition is pied-piped, as in *a pen with which to write*; cf. *\*a pen which to write with*).

MGbank treats restrictive relatives lacking an overt *wh* item as involving a null [wh] head which unlike its overt counterparts does not separate away from the relativized nominal or adverbial, and so does not move to a separate spec-CP position located below the relativized NP in the left periphery; instead, the [wh]+relativized item complex move as a unit first through the lower spec-CP and then onto the final spec-CP position. Like overt *wh* items, this complex must pass successive cyclically through any intermediate spec-vP and spec-CP positions along the way to the final surface position.

9 of the 13 cases which the parser missed were subject relatives and 4 were adverbial relatives. The parser only analysed 1 of the 5 instances of a relativized adverbial correctly relative to the PTB analysis, and this example is shown below.

(361) If the debts are repaid it could clear the way for Soviet bonds to be sold in the U.S.

The PTB has *way* as a relativized manner adverbial modifying *sold*, and the automatically generated MGbank tree agrees with this, although it must be said that to my ears *clear the way* is more plausibly analysed as a light verb structure taking a non-relativized *for-to* complement clause.

In 10 of the 13 cases which the parser failed to capture, the infinitival clause was misanalysed as a purposive adjoined to vP. These were parsing errors rather than supertagging errors, because the overt heads in both constructions are the same (only the null heads are different). Some of these sentences, such as 362 below, are arguably ambiguous between a relative and purposive reading, while for others, such as 363, the relative meaning is very strongly preferred.

(362) Later yesterday a Massachusetts senate committee approved a bill to allow national interstate banking by banks in the state beginning in num.

(363) The sound of bells is a net to draw people into the church he says

In two cases, the parser correctly identified a *wh*-relative, but incorrectly analysed the relativized item as an argument, rather than as an adverbial. One of these sentences is shown in 364 below.

(364) As banks' earnings were squeezed in the mid-1970s, the emphasis switched to finding ways to cut costs.

The parser incorrectly extracted *ways* from the spec-vP argument position of *cut*, rather than from an adjunct position (*ways* is here used as a manner adverbial, and the subject of *cut* should be NOC [pro-d]).

7 of the false positives for this construction were in reality instances of control. Examples are given in 365-367 below, where the incorrectly relativized head nouns are respectively *intention*, *plans* and *duty*. In these cases the supertagger was to blame, as it failed to assign the correct control categories to the verbs and nouns involved, allowing for the relativisation analysis to be entertained by the parser.

(365) The Baker proposal reasserts Washington's intention to continue playing a leading political role in the region.

(366) Campbell Soup not surprisingly doesn't have any plans to advertise in the magazine according to its spokesman.

(367) The First Amendment does not prescribe a duty upon the government to assure easy access to information for members of the press.

#### 5.6.3.9 Finite non-reduced subject relative clauses

Recall: 18/20

Precision: 18/20

This test set included 9 restrictive and 11 appositive subject relative clauses. The restrictive relatives included 7 clauses with *that* (treated here as a complementizer, not as a relative pronoun - see section 4.2.1) in the left periphery and 2 examples with *who*.

As was discussed in section 4.2.1, MGbank adopts a promotion analysis for restrictive relative clauses in which a null [relativizer] head with a -n licensee selections

the NP to be relativized as its complement, before the resulting larger NP is selected for by a wh-determiner, which is either an overt wh-word like *which* or *who* in the case of relative clauses with an overt wh phrase in their left periphery, or the null [wh] head for *that*-relatives and relatives lacking both *that* and an overt wh-head. In the case where the wh head is overt, the relativized NP then breaks away to check -n in the left periphery, with the wh determiner moving to a lower position in the left periphery to check -wh; for null [wh] heads, the wh+NP complex remains as a single unit, moving first to check -wh and then to check -n. In both cases, -wh movement must proceed successive cyclically through all intervening spec-vP and spec-CP positions.

For relative clauses with overt wh-items, the MGbank grammar makes a distinction between restrictive relative clauses, which restrict the set of referents for the head noun (*the Swedes who are rich hang out in Stureplan*, where it is only the rich Swedes (not the poor ones) who hang out in Stureplan) and appositive relative clauses which modify, but do not restrict the set of referents of, the head noun (*the Swedes, who are rich, hang out in Stureplan*, where all Swedes are both rich and hanging out in Stureplan). While restrictive relative clauses are plausibly analysed as complements of a determiner, this seems less appropriate for appositive relatives, which are usually offset intonationally by a brief pause (often represented in the PTB by a comma) following the relativized head noun. For this reason, MGbank adopts an adjunction-based analysis for this latter type of clause similar to the analysis adopted for relative clauses more generally in Radford (2004).

Essentially, restrictive and appositive relatives in MGbank are identical up to the null [rel] head which attracts the wh-phrase to its specifier. Whereas the resulting CP is then selected by the null [nom] head that has a +N feature that attracts the relativized NP in the case of restrictive relative clauses, appositive relative clauses lack this [nom] layer and the head noun is not generated inside the relative clause at all (so the [relativizer] head is not needed). Instead, once the [rel] head has attracted a wh-pronoun such as *who* or *which*<sup>9</sup> (this pronoun having being base generated in the relevant argument position inside the relative clause), the resulting CP is simply adjoined to the head NP which it modifies (some additional semantic composition would then be needed to ensure that the wh-word inside the relative clause and the head noun are co-indexed).

---

<sup>9</sup>Note an additional difference between appositive and restrictive relative clauses here: in restrictive relative clauses the wh-word is necessarily a determiner, because it must select the relativized NP as its complement, whereas in the case of appositives there is no NP to select and so the wh-word must be a pronoun (except for possessives such as *whose cat*), which is still of category D, but does not select a complement.

The above precision and recall scores do not differentiate between restrictive and appositive relative clauses unless *that* was misanalysed as a relative pronoun, which occurred once in the following example.

- (368) A House-Senate conference approved major portions of a package for more than \$500 million in economic aid for Poland that relies heavily on \$240 million in credit and loan guarantees.

The reason for not distinguishing in general between appositive and restrictive relatives is that the distinction is a somewhat theory internal one as far as the syntax is concerned (CCGbank does not make this distinction in the syntax, for example). To be considered correct, the *wh*-phrase (and where the clause was analysed as restrictive, also the relativized NP) had to be extracted from the correct base generated position, passing successive cyclically through any intermediate spec-vP and spec-CP positions and then landing in the correct surface position(s) in spec-CP. If the relative clause was appositive, it had to be adjoined to the correct NP. For both restrictive and appositive relatives, the entire NP that was relativized (whether by movement or adjunction) in the PTB had to be relativized in MGbank. For example, in the following sentence from the test set

- (369) Probably the most egregious example is a proviso in the appropriations bill for the executive office that prevents the president's Office of Management and Budget from subjecting agricultural marketing orders to any cost-benefit scrutiny.

the parser relativized *executive office* rather than the larger NP *the appropriations bill for the executive office*, and so this was counted as incorrect.

Both of the restrictive relative clauses with *who* were misanalysed as appositive relatives by the parser. This is because there was no information available to the supertagger to allow it to decide between tagging *who* as a determiner (for the restrictive analysis) and as a pronoun (for the appositive analysis). Conversely, in 2 cases the parser misanalysed an appositive relative as a restrictive one. As noted, however, these minor errors are not reflected in the above precision and recall scores. They could potentially be fixed by allowing the system to make reference to the comma which is found in the PTB between the head noun and the relative clause in the case of appositives.



### 5.6.3.10 Finite restrictive reduced passive subject relative clauses

Recall: 13/20

Precision: 13/14

The restrictive relative clauses evaluated in this section were those whose verbs are in the passive voice but which lack any overt passive auxiliary or any *wh*-phrase or *that* complementizer in their left periphery. An example from the test set which the parser correctly recovered is the *sought* clause in the following sentence.

(370) The final vote came after the House rejected Republican efforts to weaken the bill and approved two amendments sought by organized labor.

In MGbank, these examples are handled similarly to other restrictive relative clauses except that there is a special T head which selects a passive vP directly, rather than selecting for a voiceP. An RREL selectional feature is included on the t feature of this T head, and the complementizer *that* has a -RREL feature on its t= selector which ensures it can never select for a reduced relative TP complement, thus correctly blocking a relative clause analysis for *\*two amendments that sought by organized labor*. There is also a special [rel] head for reduced relatives, which is the only [rel] head to select for a RREL TP complement; this [rel] head has a disjunctive feature +[OP|VMOD] on its +WH licenser which ensures that the *wh*-phrase in the left periphery must either be null, as in the case of argument *wh*-phrases (thereby blocking *\*two amendments which sought by organized labor*) or adverbial (thus allowing *except where explicitly stated*, but blocking *\*except explicitly stated*).

6 of the 7 cases which the parser missed were misanalysed as involving a post-nominal adjectival phrase rather than a relative clause (this error was also noted in section 5.6.3.1 above), which is clearly a supertagging error (i.e. the passive verb was tagged as an adjective). This is not an unreasonable analysis and is in fact very close to the CCGbank analysis of these structures.

The final missed instance involved a parsing attachment error, in which only a subpart of the correct head NP was relativized. This example is shown below.

(371) The Herald joins the Baltimore News-American which folded and the Boston Herald-American which was sold as cornerstones of the old Hearst newspaper empire abandoned by the company in the 1980s.

Instead of relativising *cornerstones of the old Hearst newspaper empire*, the parser relativised the smaller NP *old Hearst newspaper empire*.

The only false positive for this construction involved a curious supertagging error in which the adverb *instead* was tagged as a passive participle verb. The example is shown below.

- (372) Could rising volatility possibly be related to uncertainty about the economics of stocks instead of the evil deeds of program-trading goblins.

#### 5.6.3.11 Free subject relatives

Recall: 19/20

Precision: 19/20

Nominal free relative clauses in MGBank are very similar to appositive relatives, except that instead of being adjoined to a head noun, they are selected for by a null [det] head which transforms them into DPs which can be selected as arguments by verbs and prepositions.<sup>10</sup> To be judged correct, the correct DP had to be relativized and the relative clause had to appear in the correct argument position of the matrix clause.

The one example which the parser missed is shown below.

- (373) The women indicated which family member usually did various household chores.

The supertagger mistakenly tagged *which* as a pronoun rather than a determiner, and then analysed *family* and *member* as DPs adjoined to the DP headed by *which*. Other than that, the construction was correct, with the large DP *which family member* correctly extracted from the subject position to spec-CP and the relative clause appearing in the correct subject argument position of the larger clause containing it.

The parser also predicted one false positive, where the clause in question was in fact an appositive relative clause. This example is given below.

- (374) Other winners Monday included nonferrous metals which attracted investors because of a surge in gold prices on the back of the unstable dollar

The supertagger mistakenly assigned the category for a number or post-determiner quantifier like *many* to *nonferrous* (marked as an adjective in the PTB), and then adjoined the resulting QP to the VP headed by *included*, with the *which* clause then

<sup>10</sup>Adverbial free relatives, such as the *where* clause in *Areas of the factory were particularly dusty where the crocidolite was used*, are not selected by the [det] head, but instead are selected by an [adjunctizer] head and then adjoined to the relevant verbal projection.

taken as the object argument of that verb (QPs headed by numbers sometimes appear as adjuncts to VPs in MGbank, an example being *4.3%* in the sentence *Government construction spending rose 4.3% to \$88 num billion* (one piece of evidence that *4.3%* is an adjunct rather than an argument here is the fact that it can be replaced by a degree adverb such as *slightly*).

### 5.6.3.12 Finite object relative clauses

Recall: 19/20

Precision: 19/19

As with finite subject relatives, finite object relatives in the test set included both appositive and restrictive relatives. There were 2 appositive relatives and 18 restrictive relatives in this test set. Of the restrictive relatives, 13 were bare in the sense that they lacked either the complementizer *that* or a wh-phrase in their left periphery. An example which the parser correctly analysed is shown below.

(375) “Oh you’re in the paper business” is one reaction Mr. Sigler says he ’s gotten from his big institutional shareholders.

Here, the DP *reaction* (headed by a null [wh] head) was correctly relativised from the object position of *gotten* and thence across the higher *said* clause to spec-CP of this clause, moving successive cyclically via both the intermediate spec-vP positions of the two clauses and the intermediate CP of the *gotten* clause.

Three of the restrictive relatives were *that*-relatives and in all three cases *that* correctly appeared as a complementizer rather than as a relative pronoun in these examples. The other two restrictive relatives involved wh-phrases with pied-piped prepositions. An example which the parser correctly analysed is shown below.

(376) Today is not the time to signal that Congress in any way sanctions the dismal state into which antitrust enforcement has fallen Mr. Edwards argued.

The other case was the one finite object relative which the parser misanalysed. The sentence is given below.

(377) The speed with which such program trades take place and the volatile price movements they can cause are what program trading critics profess to despise.

The error here was a supertagging one: the phrase *speed with which* (initially generated as the PP *with which speed*) was correctly relativised to the left periphery of the

relative clause, but it was not extracted from the correct base position as an adjunct of *take place*. Instead, it was base generated as a complement of the noun *place*, which was mistakenly assigned the category of a noun in need of a complement (*take* was also assigned the category of a regular transitive verb base generated in V, rather than of a light verb base-generated in v. This is because there is simply not enough training data for light verbs in the hand crafted seed set on which the supertagger was trained).

#### 5.6.3.13 Free object relatives

Recall: 19/20

Precision: 19/19

All of the object relative clauses were correctly analysed except the following example, which involves ellipsis of *contribute* following *can*.

(378) “Please contribute what you can,” the ad said.

The parser correctly analysed this example in every way (including predicting an imperative clause and focus movement of the quotative clause to the left periphery of the matrix clause), except that instead of generating *can* in the mod position with a null [pro-v] head as the clause’s verbal heart, it instead generated *can* in V. This is no doubt due to the fact that null pronominal heads are held back from the chart (as described in section 4.7.4) unless the parser cannot find a parse, and clearly the parser had assigned a high enough probability to the transitive category for *can* to allow it into the chart to bleed the correct analysis.



## **Chapter 6**

# **Wide-coverage neural A\* parsing for Minimalist Grammars**

## 6.1 Introduction

This chapter<sup>1</sup> presents the first ever wide-coverage parser for the MG formalism. The parser is equipped with the deep and highly constrained, wide-coverage grammar presented in chapters 3 and 4, and in Appendix A. The constraints of the grammar are hard, but many of the constraints in operation during parsing are soft and therefore best handled probabilistically.

For example, in *she cut the butter with a knife*, it is intuitively obvious that the PP *with the knife* does not modify the noun *butter*, but rather that it modifies the verb *knife*, yet clearly in other cases this situation is reversed, as in *she saw a chef with a knife*; and there are still other cases where either VP or NP attachment seems possible, as in *she saw the man with the telescope*. To make such attachment decisions, we humans rely on contextual cues and vast amounts of world knowledge to which machines do not have direct access. However, a statistical model trained on a large corpora of annotated data, such as the MG treebank presented in the previous chapter, can serve as a proxy for such world knowledge: if the system sees PPs semantically headed by *knife* modifying VPs headed by *cut* (or other verbs which are close to *cut* in vector space (such as *slice*, *chop* etc) during training, then it can potentially learn to favour VP attachment over NP attachment when it encounters a similar situation at test time.

For this reason, in addition to its formal grammar, the parser presented here also uses an adaptation of the highly efficient A\* parser currently used for CCG (Lewis and Steedman, 2014a; Lewis et al., 2016). This model is factored only over the lexical probabilities assigned by a statistical supertagger, with no model of the derivation itself at all. As Lewis and Steedman (2014a) note, simpler models are very easy to implement, replicate and extend, making this algorithm an attractive choice for a first attempt at wide-coverage statistical MG parsing.

However, porting this A\* algorithm to the MG formalism was not trivial because although CCG and MG share much in common (for example, they are both strongly lexicalised),<sup>2</sup> they also differ fundamentally in at least two important respects, namely that only MGs allow for null heads and movement operations. These differences necessitated some adaptations to the model and to the operations of the parser, and these are also described in this chapter. Also reported here are a number of experiments designed to evaluate the parser's current performance on the recovery of various types of

---

<sup>1</sup>The work presented in this chapter is based on Torr et al. (2019)

<sup>2</sup>See Berwick and Epstein (1995) on the convergence of Minimalist syntax and Categorical Grammar.

dependencies. These include global word-word (syntactic and semantic) dependencies (both labelled/directed, and unlabelled/undirected), along with unbounded object extraction dependencies. In order to assess how the parser currently performs relative to other existing parsers, comparative evaluations with a near state-of-the-art CCG parser trained on the same data and using the same A\* search algorithm are also presented.

## 6.2 A\* CCG parsing

Combinatory Categorical Grammar (CCG; Steedman 2000) is another linguistically expressive formalism capable of recovering unbounded long distance dependencies. Like MG, CCG is strongly lexicalised, with a large lexical category set and a small set of abstract combinatory rules, the most basic of which is forward/backward application (equivalent to MG's Merge). Categories are either basic (NP, S, etc) or functional. The functional categories determine the subcategorization frame of the words they label. For example, the category for a transitive verb is  $(S \backslash NP) / NP$ , which says that this word must combine with an (object) NP on its right (indicated by the forward slash), which will yield a category which must combine with a second (subject) NP on its left (indicated by the backward slash). In place of movement, CCG uses type raising and function composition rules to capture unbounded long distance dependencies.

CCG already has a very well-established research tradition in wide-coverage parsing (see, e.g., Hockenmaier and Steedman 2002b; Clark and Curran 2007b; Lewis and Steedman 2014a; Xu 2016; Lewis et al. 2016; Wu et al. 2017). A key advancement in CCG parsing that enabled it to become efficient enough to support large-scale NLP tasks was the introduction of Markovian supertagging techniques by Clark and Curran (2007b) that were originally proposed for LTAG (Bangalore and Joshi, 1999). Supertagging is essentially just part-of-speech tagging for strongly lexicalised formalisms with very large tagsets. Because these supertags contain a great deal of subcategorization information, supertagging has been described as 'almost parsing' (Bangalore and Joshi, 1999).

Inspired by the A\* algorithm for PCFGs of Klein and Manning (2003), Lewis and Steedman (L&S; 2014a) present a simple yet highly effective CCG parsing model which is factored over the probabilities assigned by the lexical supertagger alone, with no explicit model of the derivation at all. This approach is extremely efficient and avoids the need for pruning the search space, which L&S note negatively impacted the performance of earlier CKY CCG parsers. Instead, the parser considers the complete



distribution of 425 CCG lexical categories for each word.<sup>3</sup> The supertagger was originally a log linear classifier, but Lewis et al. (2016) greatly enhanced its accuracy by exchanging this for a stacked bi-LSTM neural model.

The key difference between A\* and CKY CCG parsing is the fact that A\* uses search heuristics that avoid building the whole chart without compromising the correctness guarantees. This is achieved using an agenda implemented as a priority queue of items ranked by their *cost*, calculated as a product of their *inside cost* and an upper bound on their expected *outside cost*. The agenda is initialised with the full set of 425 supertags for each word. The parser pops the item with the lowest cost from the agenda, stores it in the chart if it is not already there, and attempts to combine it with other items already present in the chart. Newly created items have their costs calculated before being added to the priority queue agenda. The entire process is repeated until a complete parse for the sentence is returned. The algorithm guarantees that the first parse returned is the most probable (i.e. the Viterbi parse) according to the model.

L&S treat a CCG parse  $y$  as a list of lexical categories  $c_0 \dots c_{n-1}$  together with a derivation, and make the simplifying assumptions that all derivations licensed by the grammar are equally likely, and that the probability of a given lexical category assignment is conditionally independent of all the other assignments given the sentence. Let  $\mathcal{Y}$  be the set of all derivations licensed by the grammar; then the optimal parse  $\hat{y}$  for a given sentence  $S$  with words  $w_0 \dots w_{n-1}$  is given as:

$$\hat{y} = \operatorname{argmax}_{y \in \mathcal{Y}} \prod_{i=0}^{n-1} p(c_i | S) \quad (6.1)$$

Let  $\alpha$  be a set of indices  $\{i, \dots, j\}$  for words  $w_i \dots w_j$  labelled with category sequence  $c_i \dots c_j$  inside some expression. The inside probability of  $\alpha$  is simply the product of the probabilities of the lexical category assignments given the sentence.

$$s(\alpha) = \prod_{i \in \alpha} p(c_i | S) \quad (6.2)$$

The upper bound estimate for the outside probability of a span  $\alpha$  is given by

$$h(\alpha) = \prod_{i \notin \alpha} \max_{c_i} p(c_i | S) \quad (6.3)$$

---

<sup>3</sup>There are 1,285 CCG lexical categories in sections 02-21 of CCGbank which are used for training CCG parsers. However, following Clark and Curran (2007b), L&S apply a cutoff such that only those lexical categories appearing 10 or more times in the training data are used for tagging, which results in a tagset of 425 items.

where  $\max_{c_i} p(c_i | S)$  is the probability of the most likely category assigned to word  $w_i$  according to the supertagger, which can be precomputed for the sentence and cached. To avoid numerical errors caused by multiplying together very small numbers, we convert the probabilities to log space *costs* and use addition rather than multiplication.

### 6.3 From CKY MG parsing to A\* MG parsing

Just as current CCG A\* parsers can be viewed as extensions of CKY CCG parsers, the same is true of the relation between the A\* MG parser constructed for this project and the CKY MG parser of Harkema (2001). In this section, we will therefore look at the operations of the re-implemented (and slight adapted) version of Harkema's CKY parser which was used within the Autobank system to semi-automatically generate MGbank (see section 5.3.2), and how this algorithm was modified in order to convert it to an A\* MG parser.

Like the classical CKY algorithm for CFGs, Harkema's bottom up CKY MG parser uses dynamic programming in the form of a chart of unique items enabling the system to avoid applying the same rule of inference to the same item multiple times. In addition to the chart of standard CKY, there is also an agenda which holds items which have yet to be added to the chart. The deductive procedure of the CKY MG recogniser is as follows (the following is adapted slightly from Harkema (2001, page 96)).

1. Initialise the chart to the empty set of items and the agenda to the axioms (i.e. the lexical items) of the deduction system.
2. Repeat the following until the agenda is exhausted:
  - (a) Select an item from the agenda, called the trigger item, and remove it from the agenda.
  - (b) If any Move operations can be applied to the trigger item, then apply these operations and add all the newly resulting items to the agenda. Otherwise, add the trigger item to the chart, if the item is not already in the chart.
  - (c) If the trigger item was added to the chart in the previous step, generate all items that can be derived from the trigger item and any items in the

chart by one application of a rule of inference (i.e. a Merge or Adjoin operation), and add these generated items to the agenda.

3. If a goal item is in the chart, the goal is proved, i.e., the string is recognised, otherwise it is not.

Converting this recogniser to a parser is a simple matter of keeping pointers from each generated expression to all its possible immediate derivational pasts; each past of an expression  $e$  consists of a single expression if  $e$  was formed from a Move operation, or a pair of expressions if  $e$  was formed by a Merge/Adjoin operation. The main difference between MG CKY parsing and CFG (and CCG) CKY parsing is that in the latter case expressions define only a single span and are merged only with other expressions which are string adjacent to them. In MG CKY parsing this is not the case: expressions may define multiple spans and the parser must in principle consider merging all items in the chart with all other items in the chart.<sup>4</sup> This is because items which are initially merged together may subsequently move away from one another and hence may not appear adjacent to one another in the surface string. Note that owing to the inclusion of the across-the-board movement mechanisms introduced in sections 3.3.7.2 and 3.3.7.4, the parser must even consider merging items with overlapping subspans.

To convert this CKY MG parsing algorithm to an A\* parsing algorithm we must make two changes. First, the agenda, which is simply an unsorted stack or queue in the CKY MG parser, must be re-implemented as a priority queue (implemented here as a Fibonacci heap), with items ranked according to their probability (or cost), with more probable (or less costly) items being selected from the agenda before less probable (or more costly) ones. Second, because in the case of A\* parsing we are only attempting to construct the Viterbi parse rather than the complete chart, it is only necessary to retain pointers from each expression to its single most probable derivational past according to the model. The basic algorithm is shown in pseudocode on the next page.

## 6.4 Porting L&S's CCG parsing model to MG

The simplicity, speed and performance of L&S's A\* CCG parser makes it an attractive option for a first implementation of a wide-coverage MG parser. However, while CCG

---

<sup>4</sup>In practice it is possible to avoid this by organising each cell of the chart according to the active feature of the head chain of each expression, so that, for instance, the parser will only consider items with =d/d= features if the trigger item's active feature is d/D.

---

**Algorithm 2** A\* MG Parsing algorithm.

---

```
1: while agenda is not empty do
2:   item1  $\leftarrow$  deleteMax(agenda)
3:   if item1 is goal item then
4:     return item1
5:   else if item1  $\notin$  chart then
6:     add(chart, item1)
7:     R  $\leftarrow$  []
8:     if can move item1 then
9:       add(R, move(item1))
10:    for item2  $\in$  chart do
11:      if can merge item1 and item2 then
12:        add(R, merge(item1, item2))
13:    for item  $\in$  R do
14:      if item  $\notin$  {chart  $\cup$  agenda} then
15:        add(agenda, item)
16:      else if item  $\in$  agenda then
17:        updateWeight(agenda, item)
```

---

and MG are similar in some respects (such as the fact that they are both strongly lexicalised), there are also some fundamental differences between the formalisms which mean that some adaptations are needed in order to port the simple model described in section 6.2 to the MG formalism.

### 6.4.1 The problem of multiple spans

The first (trivial) issue is that as we saw in chapter 3, unlike CCG expressions, MG expressions contain discontinuous spans in order to allow for movement operations. Therefore, we must redefine  $\alpha$  in Equations 6.2 and 6.3 to be the set of word indices covered by *all* the spans contained within an MG expression.

### 6.4.2 The problem of across-the-board movement

The second issue is that, as we saw in sections 3.3.7.2 and 3.3.7.4, the MGbank grammar allows for so-called across-the-board (head) movements in order to capture adjunct control, parasitic gaps, argument cluster coordination, right node raising, and other ATB coordinate structures. Consider example 379 below, for example, which features across-the-board phrasal movement of *who* out of two conjoined clauses.

(379) Who<sub>*i*</sub> did Jack say Mary likes *t<sub>*i*</sub>* and Pete hates *t<sub>*i*</sub>*?

Recall that under the MGbank analysis of examples like this, there are initially two instances of *who* in the derivation, one inside each conjunct, and that these two instances are unified at the point at which the leftmost (specifier) conjunct is merged into the main structure. Only one of those movers must contribute its cost to the cost of the resulting coordinate expression in order to avoid excessive penalisation for what is in reality just a single instance of the moving item in the surface string. The same consideration applies to cases involving ATB head movement, where in this case it is the head strings of the head chains of the two merged expressions which are unified: only one of these heads must contribute its cost to the cost of the resulting expression. We can achieve the desired effect for both ATB head and phrasal movement by first calculating the sum of the costs of the two expressions that are Merged, and then subtracting from this the cost of one member of each pair of unified movers.

In the MGbank grammar (in contrast to Kobele (2008)), it can be the case that the two unified (head) movers have different derivational histories, meaning that they may well have different costs. Example 183 of section 4.2.4.1 featured the binding of a

reflexive inside an adjunct clause which required the unification of two movers with different derivational paths. Another example, which is taken from the PTB, is given in 380 below.

(380) [<sub>TP</sub> Bond mutual funds [<sub>T'</sub> [<sub>T'</sub> offer diversification] and [<sub>T'</sub> are easy to buy and sell]]].<sup>5</sup>

What is interesting about this example is that it features a surface subject *bond mutual funds* which has been extracted in ATB fashion from two very different conjuncts. In particular, only the second bracketed conjunct in 380 contains a tough adjective, *easy*, which triggers so-called *tough movement*. In order to understand why it is necessary to unify two movers with different derivational histories here, we will now look at the MGBank analysis of tough movement.

#### 6.4.2.1 An excursus on tough movement

Tough movement is linguistically interesting because it appears to involve a DP licensed in two case (and arguably two theta<sup>6</sup>) positions, despite the fact that in general a

<sup>5</sup>Note that, as indicated, this example appears to feature coordination of T' constituents: the second conjunct contains a finite auxiliary which is standardly assumed to be situated in the T position, and both conjuncts clearly share the same subject, with surface subjects standardly assumed to reside in spec-TP. The MGBank formalism does not allow for the coordination of X' level constituents (in general accordance with the invisibility of X' nodes standardly assumed in MP). In MGBank, constructions such as this are currently simulated by including special heads whose selectee is *tbar* rather than *t*, and which lack the +CASE licenser attracting a subject. The *tbar*P's which these categories project in the phrase structure can then be coordinated, with the resulting coordinate *tbar*P complex selected as the complement of a special T head which carries the +CASE licenser which attracts a single unified subject. This is essentially a TP-shell analysis of such constructions.

<sup>6</sup>It is often argued that in fact a tough adjective does not assign a theta role to its subject. This is evidenced by the fact that the subject position can be occupied by an expletive pronoun, in a similar fashion to what we find for raising predicates.

- (1) It is easy to sell bond mutual funds.

This was the position taken in Chomsky (1981), for example. However, other authors have argued that the subject does receive a theta role from the tough adjective itself (Lasnik and Fiengo, 1974; Jacobson, 1992; Pollard and Sag, 1994; Kim, 1995; Clark, 2000; Hornstein, 2001). In the MGBank analysis of copula+adjP/PP/DP complexes, the copula is responsible for assigning the theta role to the subject, though this is currently being revised to an analysis (following Mikkelsen 2005) where the subject is generated as the specifier of a null [*prd*] head which selects the adjP/PP/DP as its complement, with the resulting *prd*P being selected for by the copula. In either case, the subject does receive an external theta role. However, the fact that expletive *it* can appear in the subject position in 1 is unproblematic in MGBank as this element is (in non-impersonal passive constructions) analysed as the residue of the extraposition (i.e. rightward movement) of a clause away from a subject position; since the extraposed clause is itself a thematic argument, the presence of expletive *it* is therefore not indicative of the absence of an external theta role. In other words, 1 is derivationally related to 2 below.

- (2) To sell bond mutual funds is easy.

DP is frozen in place for the purposes of A-movement once it has had its -case licensee checked. Another interesting property of tough movement is that it also arguably features so-called *improper movement*, in which an A'-movement step feeds subsequent A-movement. For instance, in 380, *bond mutual funds* appears in a nominative subject position in the surface string but is semantically the object of the accusative-case-assigning transitive verbs *buy* and *sell* inside the second conjunct,<sup>7</sup> while the infinitival clause embedded under the tough adjective has generally been assumed since Chomsky (1977) to be a type of infinitival relative clause with a null operator in its left periphery that has moved from the object position and is co-indexed with the overt DP subject in the higher clause<sup>8</sup> (the subject of the infinitival is usually assumed to be NOC PRO). Putting all of this together, the simplified example *bonds are easy to sell*, would have the following schematic structure (where  $A^\theta$  indicates a theta feature checking position and  $A^\phi$  indicates a case/agreement feature checking position).

(381)  $[_{TP} \text{Bonds}_i^{A^\phi(nom)} \text{ are}_j [_{VP} t_i^{A^\theta} t_j \text{ easy } [_{CP} \text{OP}_i^{A'} \text{ to } [_{VP} \text{PRO sell}_k [_{VP} t_i^{A^\phi(acc)} t_k t_i^{A^\theta} ]]]]]]$

In Chomsky (1981), the co-indexation between the overt DP subject and the null operator in the embedded spec-CP is achieved via binding. However, as we saw in sections 3.3.6 and 4.2.4 on control and reflexive binding, in the MGBank formalism, the only way to establish long-distance dependencies in the syntax is via movement. Therefore, in the MGBank analysis of 386, all the co-indexed  $i$  positions, including  $\text{OP}_i^{A'}$ , form a single chain of movement steps, meaning that  $\text{OP}_i^{A'}$  is in fact a trace  $t_i^{A'}$ . In general, movement licensees are ordered on MGBank categories in such a way that A'-movements (such as wh-movement and topicalization) follow A-movements for control and raising, thus enforcing the general ban on improper movement. In order to allow for improper movement in the case of tough movement constructions, however, MGBank uses the following null [op] head (simplified here by removing some of its fine-grained selectional features), which has the effect of a unary type-changing rule that maps an ordinary DP into a DP with additional A- and A'-movement licensees triggering the tough movement steps.

$[\text{op}] :: d\{-\text{OP}.x\} = +\text{case}\{y\} D\{\text{OP}.x\} -\text{case}\{\text{ACC}\} -\text{wh} -\text{tough} D\{x\} -\text{case}\{y\}$

<sup>7</sup>The verbs *buy* and *sell* are conjoined in the MGBank tree using the approach to lexical head coordination described in section 3.3.7.5, and *bond mutual funds* moves out of this smaller coordinate complex also via ATB movement.

<sup>8</sup>This null operator is in fact included in the original PTB tree, though the PTB does not provide an explicit formalism capable of recovering it and these items are therefore simply ignored by treebank parsers.

This category adds additional -wh, D and -case (and -tough) features allowing the tough mover to first move to the left periphery of the infinitival relative clause, before moving to pick up the external theta role of the tough predicate, and finally to check case in the surface subject position. Notice that the first -case licensee of [op] is pre-specified for ACC rather than inheriting the morphosyntactic case of the complement DP. This is to prevent tough movement from nominative case positions (for DPs syncretised for NOM and ACC case), which the following contrast shows is disallowed in English.

- (382) a. \*Jack<sub>i</sub> is tough that t<sub>i</sub> is pleased.  
 b. Jack<sub>i</sub> is tough to please t<sub>i</sub>.

At the same time, we do not want to stipulate that the DP complement of [op] obligatorily has an ACC feature (by including +ACC on the +case licenser of [op]), because nominative personal pronouns can serve as tough movers, as the following example illustrates.

- (383) He<sub>i</sub> seems tough to please t<sub>i</sub>.

Furthermore, while the lower case position of tough movement is always accusative in English, the higher case position is not always nominative. This fact is exemplified by the following ECM example.

- (384) Mary expects him<sub>i</sub> to be tough to please t<sub>i</sub>.

For this reason, the [op] head's second -case feature inherits whatever case its DP complement has (via the  $\gamma$  selectional variable, which the first -case licensee lacks). The -tough licensee is used to ensure that only tough adjectives license tough movement, as only these items bear a +TOUGH licenser (the somewhat ad hoc features +TOUGH/-tough licenser/licensee features could potentially be eliminated, with fine-grained selectional restrictions used instead). The category of a tough adjective, such as *easy*, *hard* or *tough*, is given below (see Appendix A for the full reified categories involved in tough movement as well as for details on the so-called *violin-sonata paradox*).

easy :: c{+RELAT}= +TOUGH adj

The (simplified) derived Xbar and MG derivation trees for the phrase *bonds are easy to sell* under the MGbank analysis of tough constructions are given in figures 6.1 and 6.2.



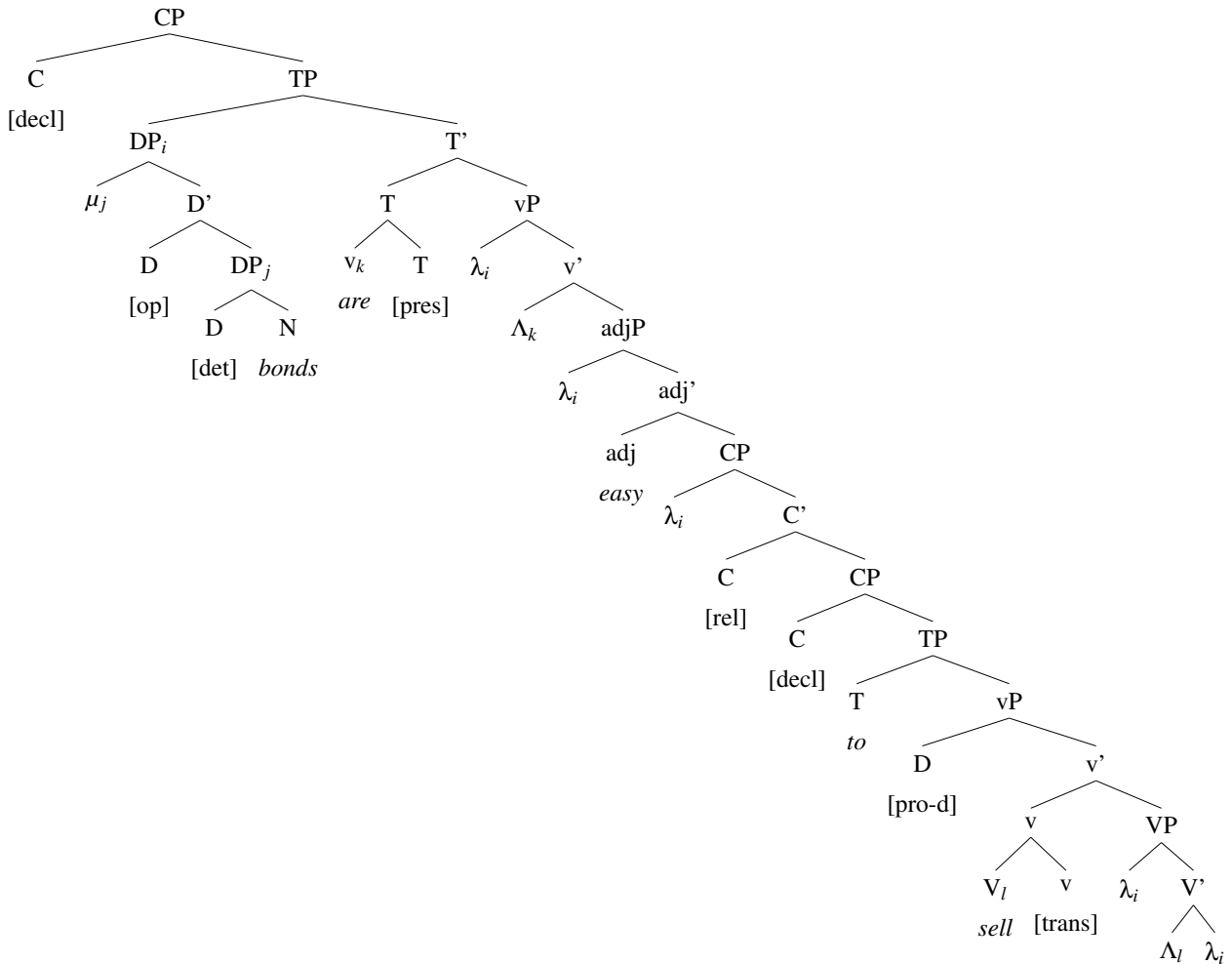


Figure 6.1: Derived Xbar tree for the sentence *bonds are easy to sell* exemplifying the MGBank analysis of tough movement (successive cyclic wh movements through spec-vP and spec of [decl] CP (see section 4.3.4) of the infinitival relative clause are omitted here to simplify the tree, but are present in the actual MGBank trees for tough movement constructions).

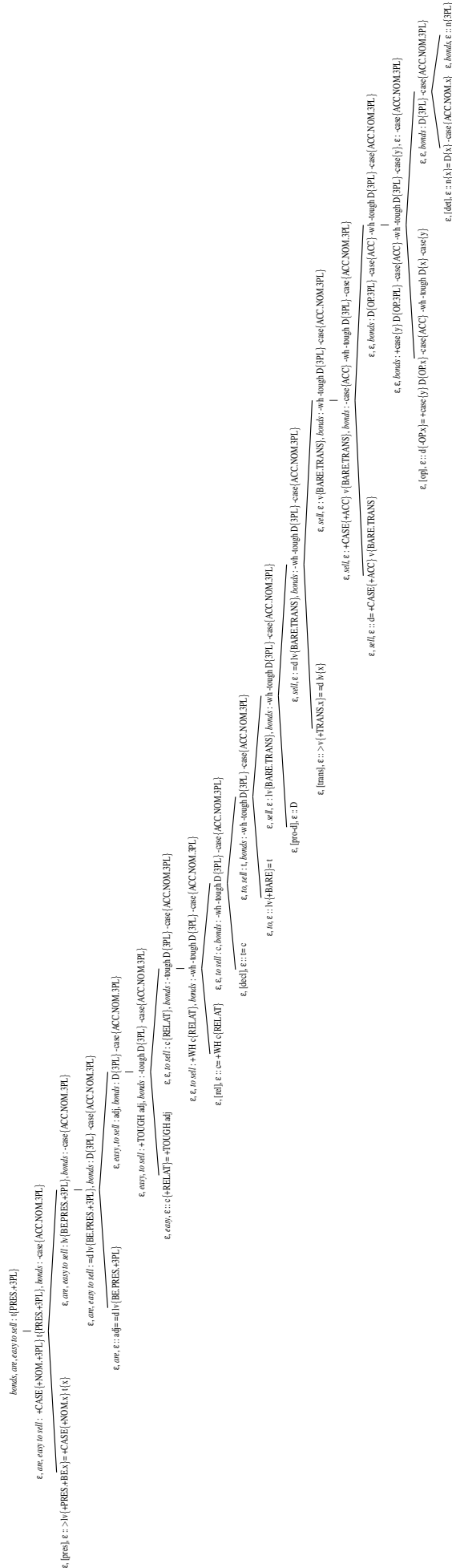


Figure 6.2: An MG derivation tree sentence *bonds* are easy to *sell* exemplifying the MGbank analysis of tough movement. See figure 6.1 for the derived Xbar tree for this sentence. Note that the final CP layer is omitted here to save space.

Returning to example 380, the important point for current purposes, is that the rightmost conjunct contains a tough adjective, whereas the leftmost one does not. During the course of the derivation, there will be two instances of *mutual bond funds* - one inside each of the two conjuncts - which must be unified when the second specifier conjunct is merged into the main structure, *despite the fact that these two instances have different derivational histories*: only the mover inside the rightmost conjunct contains the [op] head as a subconstituent.<sup>9</sup>

For the purposes of our A\* algorithm, if two unifiable movers have different costs associated with them (because they contain different supertags), then when the two structures containing them are merged, the parser uses the greater of these two costs when calculating the inside cost of the newly formed expression. If the lower cost were subtracted instead, it may make some scores non-monotonically increasing.<sup>10</sup>

### 6.4.3 The problem of null heads

The final problem for porting the CCG A\* algorithm to MGs is the most significant and relates to the fact that, unlike CCG, MG allows for phonetically null heads as we saw in section 3.2.7. Supertaggers can of course only tag what they can see (i.e. the overt words of a sentence), and yet we would like our probabilistic model to also be defined over the null heads. This section presents a novel method for effectively factoring null heads out from MG parsing by anchoring them to overt heads inside complex LTAG-like<sup>11</sup> MG lexical categories, which we will refer to here as MG supertags. We will also see how the CKY/A\* parsing algorithm described in section 6.3 can be very straightforwardly adapted to use these MG supertags.

Consider the derivation of the simple transitive sentence *he helped him*, whose derivation tree and Xbar tree are shown in figures 6.3 and 6.4 respectively. The lexical items which appear in fig 6.4 along the spine of the CP clause are shown below.

---

<sup>9</sup>Note that for sentences involving unification of movers with different derivational pasts, it will never be sufficient for the supertagger to assign only a single supertag for each word in the sentence. In fact, when using the ‘reparse’ feature in the Autobank system described in the previous chapter, if the supertag option is selected, then reparsing of any sentences involving such unification of items with different histories will currently fail owing to the fact that the system at present only records a single MG tag for each word in any given sentence.

<sup>10</sup>Note that one drawback to only using the probability of one of the two unified instances is that the strict optimality guarantees of A\* are lost. These guarantees would be retained if both probabilities were used, but then the model would overly penalise ATB movement.

<sup>11</sup>Although the MG supertags are LTAG-like in the sense that they can be viewed as constituting elementary trees, the present formalism still uses movement operations, rather than adjunction, to capture unbounded long-distance dependencies; hence it remains transformational, in contrast to LTAG.

[decl] :: t= c  
 [past] :: lv= +CASE t  
 [trans] :: >v= =d lv  
 helped :: d= +CASE v

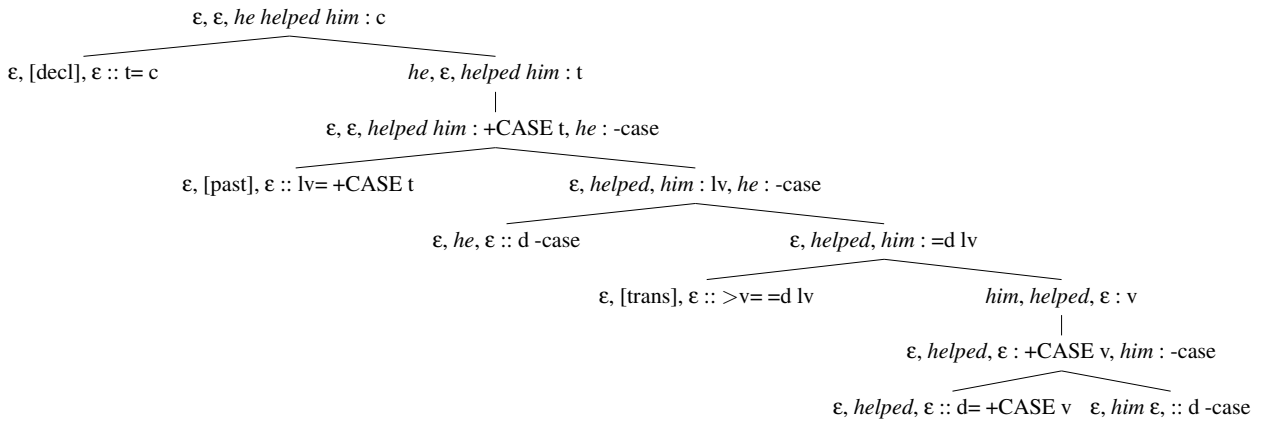


Figure 6.3: MG derivation tree for the sentence *he helped him*.

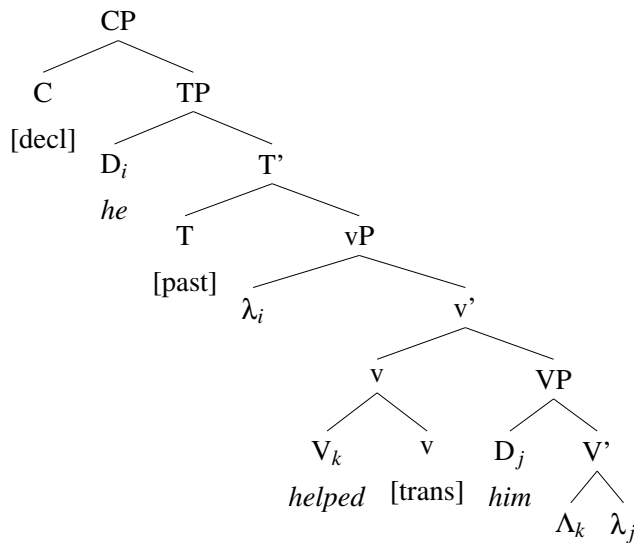


Figure 6.4: Xbar phrase structure tree for the sentence *he helped him*.

It will be useful in what follows to regard all of the phrases inside the most immediate clause containing the lexical verb, up to its CP, as being part of the *extended projection* (Grimshaw, 1991) of the lexical verb. The intuition behind this concept is that most if not all of the null heads along the spine of a clause show up as overt inflections on the verb in certain languages, hence they are in some sense part of the

projection of the verb, at least semantically.<sup>12</sup> We will also assume that nominals have an extended projection which includes DP, QP etc, running up to PP when present. The other lexical categories, adjectives and adverbs, can also be regarded as heading their own extended projections, although in most cases these extended projections trivially consist of just the adjP or advP itself. Note that many proforms, including personal pronouns, and existential and locative *there*, along with null pro-determiners such as null imperative subjects and NOC PRO (both labelled [pro-d]), also trivially constitute extended projections in their own right in MGBank.

In the derivation of our simple transitive sentence, the null [trans] little *v* Merges with the VP headed by overt *helped*, while the null [past] T head Merges with the resulting *v*P, followed by the null [decl] C Merging with the resulting TP. If we view each of these head-complement Merge operations as a head-head link in a chain, then all of these null heads are either directly (in the case of *v*) or indirectly (in the case of T and C) linked to the overt verb. All of the information represented on V, *v*, T and C heads in Minimalism is in LTAG represented on a single overt lexical category (known as an initial tree). We can adopt this perspective for Minimalist parsing (without also adopting the notions of *auxiliary tree* and *adjunction* used in the non-transformational TAG) if we view chains of Merge operations that start with some null head and end with some overt head as constituting complex overt categories. Given a corpus of derivation trees, it is possible to extract all such chains appearing in the corpus, essentially precompiling all of the attested combinations of null heads with their overt anchors into the lexicon. The pseudocode for the very simple algorithm that was used to achieve this is given below.

```

for each derivation tree  $\tau$ :
  for each null head  $\eta$  in  $\tau$ :
    if  $\eta$  is a proform:
      linkWithGovernor( $\eta$ );
    else:
      linkWithHeadOfComplement( $\eta$ );

```

<sup>12</sup>For example, we saw in section 3.2.9 that the causative little *v* head shows up as an overt causative suffix in languages like Kannada. In a full morphosyntactic Minimalist theory, the past tense *-ed* and third singular present *-s* morphemes of English would initially be generated in the T position before being suffixed either onto an auxiliary when the latter undergoes head movement to T or, in the absence of any auxiliary, onto the main verb via a type of lowering head movement known as affix hopping (Chomsky, 1957) (see Stabler (2001b) and Stanojević (2019) on incorporating affix hopping into MGs). Many Amazonian languages use suffixes on the end of the verb to mark illocutionary force; Jarawara, for example, uses a declarative suffix *-ka* (Dixon et al., 2004).

*groupLinksIntoSupertags()*

For each derivation tree, we first anchor all null heads either directly or indirectly to some overt head. This is achieved by extracting a set of links, each of which represents one Merge operation in the derivation tree. Each link is comprised of the two atomic MG lexical categories that are the lexical heads of the arguments to the Merge operation along with matching indices indicating which features are checked by the operation. Applying the algorithm to our example sentence would result in the following 3 links for the supertag anchored by the verb:

link1: [decl] :: t=<sup>1</sup> c, [past] :: lv= +CASE t<sup>1</sup>

link2: [past] :: lv=<sup>2</sup> +CASE t, [trans] :: v= =d lv<sup>2</sup>

link3: [trans] :: v=<sup>3</sup> =d lv, *helped* :: d= +CASE v<sup>3</sup>

The majority of null heads are simply linked with the head of their complement, the only exception being that null proforms are linked to whichever head selects for them (i.e. their governor). Assuming that certain null proforms (such as the [pro-v] heads used for VP ellipsis in MGbank) are the only null heads that can appear at the bottom of any extended projection,<sup>13</sup> this ensures that all of the lexical items inside a given supertag are part of the same extended projection (except for [pro-d] heads, which as noted trivially constitute extended projections in their own right but which will be contained within the supertag anchored by the overt verb of which they are an argument). This condition prevents ad hoc supertags of arbitrary structural depth from being created. Note that some atomic overt heads (such as *he* and *him* in our example sentence) will not be involved in any links and will therefore form simplex supertags.

Once the merge links and unattached overt heads are extracted, the algorithm then groups them together in such a way that any lexical items which are chained together either directly or indirectly by merge links are contained in the same group. Because links are only formed between null heads and their complements, and not between heads and specifiers (again, except in the case of [pro-d] when Merged into spec-vP) or adjuncts, and because each chain ends with the first overt head encountered, every (null

<sup>13</sup>Null verbal ellipsis heads can in fact have overt DP complements, as in *the green book which Mary hasn't read, and the blue one which she has*, which features VP ellipsis in the second conjunct with the DP object *the blue one* escaping this ellipsis via relativisation to the left periphery of the clause. This is why it is important to stipulate that null proforms must be linked with their governor rather than with the head of their complement: intuitively, the ellipsis should be conditioned on the auxiliary, *has*, not on any of the words inside the relativized nominal.

or overt) head is guaranteed to appear in just one group and each group is guaranteed to contain at most one overt lexical anchor.

The above merge links would form one group, represented compactly as follows:

$$\begin{aligned} [\text{decl}] &:: t=^1 c \\ [\text{past}] &:: lv=^2 +\text{CASE } t^1 \\ [\text{trans}] &:: v=^3 =d lv^2 \\ \text{helped} &:: d= +\text{CASE } v^3 \end{aligned}$$

To convert this reified complex lexical item to an abstract supertag category, we simply replace the phonetics of the overt anchor with the string place holder *s*, as shown below.

$$\begin{aligned} [\text{decl}] &:: t=^1 c \\ [\text{past}] &:: lv=^2 +\text{CASE } t^1 \\ [\text{trans}] &:: v=^3 =d lv^2 \\ s &:: d= +\text{CASE } v^3 \end{aligned}$$

All of the combinatorial information that was previously spread out among the different overt and null heads is now contained within this single overt supertag, which is somewhat akin to an LTAG tree fragment. There are important differences with LTAG, however. For instance, if an auxiliary verb were present between little vP and TP, then only little v would be anchored to the main verb, while T and C would be anchored to the structurally higher auxiliary.

As we saw in chapter 3, C is the head triggering A'-movements, such as wh-movement and topicalization. A consequence of this is that, although like LTAG supertags (but unlike CCG supertags), these MG supertags lexicalise A'-movement onto an overt verb, the particular verb (i.e. main or auxiliary) onto which the A'-movement is lexicalised will vary in MG, but not LTAG, depending on which other overt verbal heads are present. In many cases, the verb in question will be structurally and linearly much closer to the A'-moved element in MG than in LTAG. Furthermore, LTAG also allows for special adjunction operations to capture unbounded movements. For instance, the sentence *what did she say Pete eats for breakfast?*, would be generated from an initial tree whose string yield is *what Pete eats for breakfast* and an auxiliary tree with the yield *did she say* which is adjoined into a position between *what* and *Pete* in the initial tree. An LTAG would therefore precompile the wh-'movement' onto the supertag for *eats*, whereas in the present MG, which uses genuine movement rather than adjunction, this information would be precompiled onto *did*, the closest overt head

c-commanded by the null [int] head hosting the +WH licenser.

As noted in Kasai et al. (2017), LTAG's lexicalisation of unbounded A'-movement is one reason why supertagging has proven more difficult to apply successfully to LTAG than to CCG, Markovian supertaggers being inherently better at identifying local dependencies. Lexicalising A'-movement onto a supertag that is linearly closer to the moved item could therefore ultimately prove advantageous, in this respect. On the other hand, varying the head onto which A'-movements are lexicalised clearly necessitates a greater number of supertags than consistently lexicalising it to the same head, which increases data sparsity and is therefore likely to negatively impact supertagging accuracy.

Another example of a supertag which the above algorithm would extract, this time from the derivation tree featuring tough movement in figure 6.2, is given below.

$$\begin{aligned} [\text{op}] &:: d\{-\text{OP.x}\}=^1 +\text{case}\{y\} D\{\text{OP.x}\} -\text{case}\{\text{ACC}\} -\text{wh} -\text{tough} D\{x\} -\text{case}\{y\} \\ [\text{det}] &:: n\{x\}=^2 D\{x\}^1 -\text{case}\{\text{ACC.NOM.x}\} \\ s &:: n\{3\text{PL}\}^2 \end{aligned}$$

This is the supertag for a 3rd person plural, non-pronominal DP, which will undergo tough movement.

#### 6.4.4 Adapting an existing CKY MG parser to use MG supertags

The MG supertags can be integrated into an existing CKY MG parser (which may then form the basis of an A\* MG parser) quite straightforwardly as follows: first, for each supertag token assigned to each word in the sentence, we map the indices that indicate which features check each other into globally unique indices. This is necessary to ensure that different supertags and different instances of the same supertag assigned to different words are differentiated by the system. We also store all the obligatory Merge relations encoded by the supertags in a lookup table. Then, whenever one of the constrained features is encountered, the parser uses this lookup table to ensure that this feature is only checked against the feature with the matching index. The parser otherwise operates as usual except that thousands of potential Merge operations are now disallowed, with the result that the search space is potentially reduced drastically (though this does of course depend on how many candidate supertags it is necessary to assign to each word, which in turn depends on how accurate the supertagger is). During parsing, the overt heads carry the entire cost of their supertag into the agenda;



the null heads are simply assigned a zero cost.

One complication concerns the dynamic programming of the chart. In standard CKY MG parsing, as with classical CFG CKY, items with the same category spanning the same substring are combined into a single chart entry during parsing. This prevents the system having to create identical tree fragments multiple times. But the current approach complicates this because many items now have different predetermined futures (i.e. their unchecked features are differentially constrained), and when the system later attempts to reconstruct the trees by following the backpointers, things can become very complicated. We can avoid this issue, however, simply by treating the unique identifiers that were assigned to certain selector features as part of the category. This has the effect of splitting the categories and will, for instance, prevent two single chain categories  $d=^1 =d v$  and  $d=^2 =d v$  from being treated as a single chart entry until their  $d=$  features have been deleted.

## 6.5 Experiments

This section reports on a number of experiments which were carried out in order to evaluate the performance of the A\* MG parser described in the previous sections of this chapter.

### 6.5.1 Model description

Two types of MG grammars were used in the experiments described here: Abstract and Reified. The difference between them is that in the Abstract grammar most of the 100 or so fine-grained selectional and agreement restriction features described in section 3.4.1 have been removed with the exception of the following 5 features, which are necessary to the inner workings of the parser: ANA, EDGE, IT, +NONE, MAIN.<sup>14</sup>

<sup>14</sup>In fact ANA is now a deprecated feature which will soon be removed from the corpus entirely. It was previously used to allow for DSMC violations in certain constructions involving the binding of reflexive and reciprocal anaphors, but the analysis of these constructions has since been revised and no longer requires this hack; as was discussed in section 4.3.1.1, EDGE is used to allow certain movers to escape SpIC; IT is also used to enable violations of SpIC, this time not by a certain type of mover, but from within a certain type of specifier, specifically from DP-CP complexes headed by *it* in extraposition contexts: in MGBank, a sentence like *what is it difficult to know?* is generated from the underlying sentence [*it to know what*] *is difficult*, with the determiner *it* taking the CP *to know what* as its complement, with the resulting DP *it to know what* then being merged into spec-vP of *is* (or [prd] in the new analysis of copulas to which MGBank is being migrated). *what* then moves out of this specifier (in violation of SpIC) to the left periphery of the matrix clause, while *it to know* moves to spec-TP of that clause, before *to know* moves out of this specifier (again, in violation of SpIC; in actual fact, *to know* moves away from *it* the moment they are merged, just as *what* did) and undergoes rightward movement

The Reified grammar is more constrained, which should make it more precise (at some expense to recall) but at the same time more difficult to supertag correctly due to the sparsity that comes with a higher number of supertags. Extracting the complex MG supertags from the entire MGbank corpus results in a Reified tagset of 3926 items and an Abstract tagset of 2644 items<sup>15</sup>.

For both Abstract and Reified, the same supertagging neural architecture was used<sup>16</sup>, which works by initially embedding the word tokens using the final layer of an ELMo embedder (Peters et al., 2018), followed by a single affine transformation to compress the embeddings into a vector of size 128 for each word. These embeddings are further fed into a two layer bi-LSTM (Hochreiter and Schmidhuber, 1997; Graves, 2013). Finally, the hidden states of the final layer of the bi-LSTM are passed through a two layer MLP to predict the distribution of the supertags for each word. The parameters are trained using an Adam optimiser with a learning rate of 0.0002.

### 6.5.2 Recovery of MGBank dependencies

The parser was first tested on its ability to recover global syntactic and semantic (local and non-local) dependencies extracted from MGBank. Because this corpus only covers a little more than half of the PTB sentences, both sections 00 and 01 of MGBank were used for development and both sections 23 and 24 were used for testing; sections 02-22 were used for training.

Both labelled directed and unlabelled undirected bi-lexical dependencies were extracted for each binary non-terminal in the Xbar phrase structure trees transduced from the derivation trees and included in MGBank. See sections 5.4.4.1 and 5.6.1 for details of this extraction procedure and on the difference between the syntactic and semantic dependencies. Importantly, as was noted in section 5.6.1, the dependency evaluation used here includes the non-local dependencies generated by all the various types of movement in MGBank and is therefore considerably tougher than the purely local dependency evaluation used in Collins (1999) (and is more akin to the third method used

---

to the end of the sentence. Both of these SpIC violations are permitted owing to the IT feature on the d feature of *it* and a hack in the parser code which says that SpiC can be violated if the specifier in question has this property on its active feature; as discussed in section 4.3.5, +NONE is used to create certain hard island effects (such as the complex-NP constraint); finally, MAIN is used to identify a main clause C head. This allows the parser to avoid returning subordinate CPs as complete parses of a sentence.

<sup>15</sup>This number of tags is closer to the 4727 elementary trees of the TAG treebank of Chen (2001) than to CCGbank's (Hockenmaier and Steedman, 2007) 1286 lexical categories.

<sup>16</sup>The supertagger was coded up by Miloš Stanojević.

in Clark and Hockenmaier (2002)).<sup>17</sup> This is because the parser will be penalised for not recovering all the various long-distance dependencies which were discussed in chapters 3 and 4, and in section 6.4.2.1 of the current chapter. As an example, consider the following sentence, which was taken from section 00 of the PTB and was one of the test sentences for the relative clause evaluation which is reported in section 6.5.6.2 below.

(385) Mrs. Ward says that when the cheating was discovered, she wanted to avoid the morale-damaging public disclosure that a trial would bring.

This sentence features several types of long-range dependencies which standard dependency and context-free constituency parsers would fail to capture. For example, there are two relative clauses in this sentence, both of which feature unbounded wh-movement. The first of these is the adverbial free relative *when the cheating was discovered*, which modifies the verb *wanted*; inside this clause, the wh-adverb *when* appears in the left periphery of the clause, but clearly modifies the verb *discovered* located 4 words away; the verb is also in the passive voice, meaning that from a Minimalist perspective, the subject *the cheating* must have been base-generated as its deep object (because it is the THEME argument of *discovered*) before moving to the surface subject position; this is admittedly a more theory-internal long-range dependency and therefore not one which other parsers should be penalised for not predicting. The second relative clause is the restrictive object relative *morale damaging public disclosure that a trial would bring*, in which *morale-damaging public disclosure* has been extracted from the object position of *bring* to the left periphery of the clause. Finally, this sentence also contains a control relation between the subject of the *want* clause (*she*) and the null subject of the *avoid* clause. None of these long-range dependencies would be evaluated by a purely local dependency-based metric such as Collins (1999), but all of them are captured by the dependency evaluation used here.

The Abstract MG parser failed to return a parse for this sentence, but on its single run, the Reified parser returned a flawless parse that included the correct recovery of all the aforementioned long-range dependencies. The parse tree is too large to show in graphical form, but the Xbar bracketing which the parser produced is shown below. Note that in addition to all of the long-range dependencies noted above, Minimalist theory also predicts additional V-to-v head movements of the verbs in this sentence (traces of head movement are marked in the tree by  $\Lambda$  in contrast to  $\lambda$  for phrasal move-

<sup>17</sup>As in Collins (1999), the labels are triples of the parent, non-head child and head child categories.

ment) and successive cyclic wh-movements in the restrictive relative clause through the intermediate spec-vP position, and these were all also correctly recovered by the parser; successive cyclic movement does not proceed through spec-vP of the free relative clause, however, because the verb in this clause is passive, and the intransitive little *v* which governs it is standardly assumed not to constitute a (strong) phase (see section 4.3.4 for discussion).

(TP (DP-15 (D' (D ([det]))(NP (NP (N' (N (mrs.))))(AP (A' (A ([adjunctizer]))(NP (N' (N (ward)))))))))))(T' (T ([pres]))(vP (DP-15 ( $\lambda$ ))(v' (v (V-14 (says))(v ([trans])))(VP (V' (V-14 ( $\Lambda$ )))(CP (C' (C (that))(TP (AP (A' (A ([adjunctizer])))(CP (AP-4 (A' (A (when)))))(C' (C ([rel])))(CP (AP-4 ( $\lambda$ )))(C' (C ([decl]))(TP (TP (DP-1 (D' (D (the)))(NP (N' (N (cheating)))))))(T' (T (voice-3 (was)))(T ([past])))(voiceP (voice' (voice-3 ( $\Lambda$ )))(vP (v' (v (V-2 (discovered))(v ([intrans])))(VP (V' (V-2 ( $\Lambda$ )))(DP-1 ( $\lambda$ )))))))))))(AP-4 ( $\lambda$ )))))))(TP (DP-12 (D' (D (she)))))(T' (T ([past]))(vP (DP-12 ( $\lambda$ ))(v' (v (V-13 (wanted))(v ([trans])))(VP (V' (V-13 ( $\Lambda$ )))(CP (C' (C ([decl]))(TP (T' (T (to)))(vP (DP-12 ( $\lambda$ ))(v' (v (V-11 (avoid))(v ([trans])))(VP (DP-10 (D' (D (the)))(NP (DP-6 (NP-5 (N' (N ([relativizer])))(NP (AP (A' (A ([adjunctizer])))(NP (AP (A' (NP (N' (N (morale))))(A (-))))(NP (N' (N (damaging)))))))(NP (AP (A' (A ([adjunctizer])))(AdjP (Adj' (Adj (public)))))))(NP (N' (N (disclosure)))))))(D' (D ([wh]))(NP-5 ( $\lambda$ )))(N' (N ([nom])))(CP (DP-6 ( $\lambda$ )))(C' (C ([rel])))(CP (DP-6 ( $\lambda$ )))(C' (C (that))(TP (DP-8 (D' (D (a)))(NP (N' (N (trial)))))))(T' (T (mod-9 (would)))(T ([pres])))(modP (mod' (mod-9 ( $\Lambda$ )))(vP (DP-8 ( $\lambda$ ))(v' (DP-6 ( $\lambda$ ))(v' (v (V-7 (bring))(v ([trans])))(VP (DP-6 ( $\lambda$ ))(V' (V-7 ( $\Lambda$ ))(DP-6 ( $\lambda$ )))))))))))))))(V' (V-11 ( $\Lambda$ ))(DP-10 ( $\lambda$ ))))))))))))))))))

Table 6.1 shows the results on the MGbank test set. On both syntactic and semantic dependencies, the Reified model has higher precision, F1-score and exact matching of the whole dependency structure but, with the exception of the labelled syntactic dependencies, has a lower score on recall owing to the constraining impact of the selectional and agreement features; the Abstract model parsed 1924 sentences (96.5%) out of 1998 in the test set, while the Reified model parsed 1902 (95.4%). These F1 scores are respectable for a first attempt at wide-coverage MG parsing, although it should be pointed out that the MGbank test set is somewhat easier than the PTB test set because the sentences in MGbank are on average 4.8 words shorter than those in the PTB. The types of errors made by the parser are very similar to those which the CKY parser made during the treebanking process (many of these errors were described in detail in section 5.6.2). This suggests that improvements to the quality of MGbank will directly translate into improved scores for the parser.

		model	F1	P	R	E
syntax	LAB	Abstract	79.33	81.87	76.94	21.01
		Reified	<b>80.10</b>	<b>83.43</b>	<b>77.02</b>	<b>21.61</b>
	ULAB	Abstract	84.57	87.15	<b>82.14</b>	29.59
		Reified	<b>85.19</b>	<b>88.63</b>	82.02	<b>30.49</b>
semantics	LAB	Abstract	74.90	77.17	<b>72.75</b>	20.96
		Reified	<b>75.47</b>	<b>78.53</b>	72.64	<b>21.56</b>
	ULAB	Abstract	83.69	86.16	<b>81.36</b>	33.30
		Reified	<b>84.11</b>	<b>87.47</b>	81.01	<b>34.50</b>

Table 6.1: Results on the whole MGBank test set with P, R and E stand for precision, recall and exact match respectively. For all dependency types, precision and F1 are higher for the Reified grammar than for the Abstract one.

### 6.5.3 Comparison to CCG

Cross-formalism comparison is in general a difficult task (Clark and Curran, 2007a) because it is necessary to account both for (1) the differences in how the parsers work and (2) the differences in the kinds of structures they predict. To control for (1) a CCG parser was constructed similar to L&S’s CCG A\* algorithm but using the same supertagger as was used for the MG A\* parser to make the comparison fair. The CCG supertagger was trained on the CCG trees from CCGbank, but only on those sentences that are also present in MGBank. This CCG parser was then initially tested on its ability to recover the CCGbank dependencies for the test sentences also appearing in MGBank, and the results were compared to those of a popular, off-the-shelf CCG parser, namely EasyCCG, that was trained over the whole of the CCGbank training set. The results are shown in Table 6.2.<sup>18</sup> Our CCG parser shows much better performance in spite of being trained on much less data than EasyCCG, making it a very tough point of comparison for the A\* MG parser.

To account for (2), the CCG and MG parsers were evaluated on their ability to recall the dependencies for which both CCGbank and MGBank agree by taking as the test set the intersection of the gold unlabelled undirected CCGbank and syntactic MGBank dependencies for all sentences appearing in the MGBank test set. MGBank and CCGbank apply a different tokenization to words, and so the dependencies were compared only over the subset of sentences for which all tokens match. This turned

<sup>18</sup>Miloš Stanojević coded up the CCG parser and performed the comparative evaluation against Easy-CCG.

	model	F1	P	R	E
LAB	Our CCG A*	<b>87.4</b>	87.2	<b>87.6</b>	<b>40.0</b>
	EasyCCG A*	83.8	87.2	80.7	31.4
ULAB	Our CCG A*	<b>92.8</b>	92.5	<b>93.0</b>	<b>47.2</b>
	EasyCCG A*	90.1	<b>93.8</b>	86.8	35.9

Table 6.2: Results of CCG parsers on all 1994 sentences of MGbank test set for CCG dependencies.

out to be 1250 sentences of the MGbank test set with on average 10 dependencies per sentence that are agreed on by the CCGbank and MGbank gold standard dependency sets.<sup>19</sup> These dependencies included both local and (unbounded) non-local dependencies. Note that precision could not be computed due to the difficulties in normalising predictions on the CCG and MG sides: one might predict more dependencies which may be correct but are not predicted by the syntactic theory used in the other parser and therefore would be penalised.

The results of the evaluation are shown in Table 6.4. The CCG parser clearly gives better results, although the MG parser performs very respectably given the greater complexity of its underlying mechanisms and given that it is up against a near-state-of-art parser for a formalism with a much longer history in wide-coverage parsing. The higher performance of the CCG parser is probably largely the result of a more complete search due to the lower complexity of the formalism (the CCG parser parsed all sentences) and of the much smaller supertag set that is easier to predict as evident in Table 6.3. This inevitably means that the MG parser requires a larger amount of training data than the CCG parser to achieve similar levels of accuracy and efficiency (because the speed of A\* parsing depends on the quality of the probabilistic model). We tried replacing all MG supertags occurring less than twice in the training data with UNK tags to reduce the size of the tagset and the noise from unreliable tags, and then ignoring UNK predictions by the supertagger at test time, but this only hurt performance. Once MGbank’s coverage is increased, and more sophisticated models tailored specifically to MGs are developed, the performance gap between the formalisms should hopefully narrow.

A further consideration is that the MG parser is currently a prototype implementation in Python, hence it was necessary to prune the search space by retaining only

<sup>19</sup>Syntactic MGbank dependencies were used because this yielded more dependencies in the intersection of the CCGbank and MGbank test sets than using semantic MGbank dependencies.

top k	CCG	MG Abstract	MG Reified
1	95.73	83.11	80.62
5	99.41	97.22	95.89
10	99.64	98.42	97.66
20	99.78	99.01	98.42
40	99.83	99.26	98.81

Table 6.3: Supertagging accuracies for each grammar as the probability of having the correct supertag in the top-k predictions per word.

parser	R	E
CCG A*	95.30	69.03
MG Abstract A*	91.75	54.38
MG Reified A*	92.65	55.67

Table 6.4: Results on overlapping dependencies from gold CCG and gold syntactic MG dependencies.

the 40 most likely supertags per word. Even so, the parser still timed out on a few sentences, with a timeout setting of 30 mins. Once reimplemented more optimally in a faster language, the parser’s recall should increase as it will have more time to explore a less aggressively pruned search space.

Finally, it is important to bear in mind that the MGbank trees are also linguistically richer than their CCGbank counterparts, including several long-distance dependency types which the CCG trees lack<sup>20</sup>, and making a number of distinctions (such as between raising and control) which the CCGbank trees do not. Thus while they are more difficult to correctly recover in the first place, once recovered, they can also be more informative.

<sup>20</sup>E.g., for the binding of local reflexive and reciprocal anaphors (*himself, ourselves, their own, each other*, etc) and floating quantifiers (*all, each, both* etc), the relation between the two subconstituents of a discontinuous quotation (“*funny thing,*” *says the kicker,* “*both these candidates are named Rudolph Giuliani.*”), as well as dependencies enabling the system to constrain the distribution of polarity items such as *anymore, anyway* and *much*, and to mark the scope of correlative focus coordinators like *either, neither* etc.

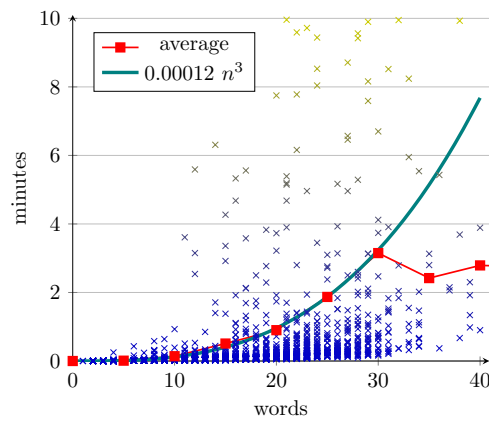


Figure 6.5: Parsing speed for Abstract model on test set.

### 6.5.4 Parsing speed

As discussed in section 4.5, the CKY MG parser of Harkema (2001), when augmented with head movement, has a worst case time complexity of  $O(n^{4k+12})$  where  $k$  is the size of the set of licensee features. We also saw in that section that owing to DSMC,  $k$  is effectively 4 in the MGBank grammar, so that the CKY parser used to generate MGBank has a worst-case complexity of  $O(n^{28})$ . The A\* MG parser described here operates in a very similar fashion to this CKY parser except that it takes an additional multiplicative cost of  $O(\log n)$  due to the usage of a heap data structure for implementing the agenda.  $O(n^{28} \log n)$  is, of course, a prohibitively high time complexity. However, although A\* does not improve on the worst case theoretical complexity of CKY, it can dramatically improve its practical expected complexity.

Figure 6.5 shows the scatter plot of parsing times for different sentence lengths and the average curve. The average curve is less informative in very long sentences due to the smaller number of parses, but in regions where there are more data points a clear pattern can be observed: a cubic polynomial curve approximates average time taken to parse sentences extremely well, which means that the expected time complexity of MG parsing with the grammar and statistical model presented in this dissertation is  $O(n^3)$ . This is much better than the worst case analysis, although the variance is high, with some sentences still requiring a very long time to parse.

Recently, Stanojević (2019) has shown that with relatively small adjustments to the parser’s inference rules, MGs with head movement can be parsed in  $O(n^{2k+5})$  time in the worst case,<sup>21</sup> which for the MGBank grammar equates to  $O(n^{13})$ , a dramatic

<sup>21</sup>Fowlie and Koller (2017) previously demonstrated that MGs *without head movement* could be parsed in  $O(n^{2k+3})$  worst case time, which was already a dramatic improvement over Harkema’s origi-



improvement over  $O(n^{28})$ . We hope to leverage these efficiency gains in the future to improve the expected time complexity of the parser. The reader may also have noticed that the absolute times in 6.5 are very slow by modern parsing standards. This is very likely due to the fact that the parser is currently implemented non-optimally in Python. As a next step, the parser will be re-implemented in a faster language, using the optimisations to the chart which are suggested in Stanojević (forthcoming); these two changes alone should improve absolute parsing speed considerably.

### 6.5.5 Coverage

Section 00 of the PTB contains 1921 sentences with an average sentence length of 21.9 words; other than a 212 word outlier, the maximum sentence length is 96. When run over all of these sentences, the Reified parser returned parses for 1490 (77.6%) sentences with an average sentence length of 14 and a maximum sentence length of 53. The Abstract parser returned 1549 parses (80.6%) with an average sentence length of 15.3 and a maximum sentence length of 49. The CCG A\* parser returned 1909 parses (99.4%).

### 6.5.6 Recovery of unbounded object extraction dependencies

As discussed in the introduction to this thesis, the recovery of unbounded dependencies, including wh-object questions, is a primary motivation for parsers based on deep grammar formalisms. Wh-object questions themselves are extremely rare in the PTB, but object relative clauses, which also involve unbounded movement, are relatively frequent. Following Clark et al. (2004), the A\* MG parser was therefore also evaluated on all the free and non-free object (and embedded subject) relative clauses in section 00 of the Penn Treebank, as well as on the two examples of tough movement.

There are 24 examples of non-free object relative dependencies across 20 sentences in section 00, and 17 free object relative dependencies across 16 sentences. All of these sentences, along with indications of which dependencies our parser did and did not recover, are given in figures 6.7 and 6.6, where they are presented using the tokenization our MG parser used. For comparison, the CCG A\* parser was also evaluated on this task but was trained and tested using the original CCGbank tokenization.

---

nal result. However, Stanojević (2019) shows that adding head movement to Fowlie and Koller’s system increases complexity to  $O(n^{2k+9})$ .

### 6.5.6.1 Recovery of free object relative dependencies

On the free object relatives, our abstract parser performed best, recovering 13/17 dependencies. The parser only predicted 14 free object relatives meaning that the precision was 13/14. Of the 4 free object relative dependencies in the data which it missed, 3 were in very long sentences on which the parser timed out, suggesting that a faster re-implementation<sup>22</sup> may achieve higher recall. In the one case which the parser actually got wrong, it correctly identified that there was a free object relative dependency, and even correctly identified the correct verb. However, the verb in question was a double object verb and the system extracted the wrong object. Clark et al. (2004) reported recall of 14/17 (with precision 14/15) on these free object dependencies, while our A\* CCG parser recovered 15.5/17 with precision 15.5/17 (we awarded the CCG parser half a point for sentence 15 because it related *what* to *thinking* but not *feeling*, which it analysed as intransitive).

### 6.5.6.2 Recovery of non-free object relative dependencies

Non-free object relatives are harder than both wh object questions and free object relatives because they require a head noun to be identified in addition to an extraction site. On a single run, the Abstract parser performed best here, retrieving 10/24; the CCG A\* parser recovered 15/24, with precision of 15/21 (Clark et al. (2004) reported recall also of 15/24 and precision of 15/20). However, by allowing the system to reparse sentences on which it failed to return a parse, each time adjusting the tag dictionary threshold<sup>23</sup> setting, the Reified parser retrieved 13/24 with precision 13/17.

In two of the errors made by the Reified parser, the system correctly identified the extraction site, but relativized the wrong NP. For example, in sentence 1, the parser attached *whom Sony hosted for a year* to *complaint* rather than to *American*. Appositive relative clauses such as this are treated as involving adjunction of the relative clause to the head noun in MGBank (in contrast to restrictive relative clauses, which as we saw

<sup>22</sup>The parser is currently implemented in Python.

<sup>23</sup>A tag dictionary lists all the categories which were seen with a word during training. For a given tag dictionary threshold,  $k$ , for all words seen at least  $k$  times, the supertagger is only permitted to assign supertags which were actually seen with that word during training (this idea is borrowed from CCG parsing (Clark and Curran, 2004)). Two tag dictionaries were used here, one for just the hand-crafted seed data, and one for the whole of MGBank, including both the hand-crafted trees and the automatically generated trees. The thresholds for the seed tag dictionary/auto tag dictionary threshold for the first two rounds were 3/5 and 10/10; for the final round, the tag dictionary was turned off completely. For the other experiments described in this chapter, no tag dictionary threshold was used and the supertagger was therefore permitted to assign any supertag to any word.

in section 4.2.1 involve movement of the head noun from the object position inside the relative clause to the left periphery of that clause). The choice of attachment to either *American* or *complaint* is underdetermined by the model here, since the same supertag containing the requisite [rel] and [adjunctizer] null heads will be assigned to *hosted* in either case. One way to resolve such ties would be to augment the simple supertag-factored model with a secondary head-dependency model; an alternative would be to hard code the constraint in the grammar using fine-grained subcategorization properties and requirements such as HUMAN and +HUMAN.

In sentence 8, which involves a restrictive relative clause, the parser incorrectly assigned the supertag containing the [relativizer] null head to the noun *esteem* rather than to *damage*, hence *self-esteem* was incorrectly analysed as the extracted object rather than *damage*. The other two errors also arose out of tagging mistakes (e.g. of *where* as a complementizer, rather than an adverbial relative pronoun) which led to the incorrect prediction of an object extraction dependency.

### 6.5.6.3 Recovery of tough movement dependencies

We also evaluated the parser on the two tough movement examples in section 00, one of which is given in 386 below (the MGBank analysis of tough movement was discussed in section 6.4.2.1).

(386) That got hard to take, he added.

The MG parser failed to correctly analyse either of the two tough movement examples in section 00 owing to supertagging errors. For example, in 386 there are three important tagging decisions to be made: *hard* must be assigned the supertag for a tough adjective; *that* must be assigned the supertag for a pronoun which undergoes tough movement; and *take* must be assigned the supertag for a transitive verb. The highest scoring tag assigned to *hard* by the Abstract supertagger was the supertag for a regular adjective that takes a CP complement (*eager to help*). The correct tough adjective supertag, meanwhile only ranked 14th, meaning that the A\* search algorithm never got to consider it. Furthermore, the highest ranked tag for *take* was the supertag for an unergative intransitive verb, with the correct transitive verb tag appearing in second place. Finally, the supertag for a pronoun undergoing tough movement was not included in the 40 tags assigned to *that* owing to the fact that this supertag did not appear in the training data at all. We tried increasing the 8 examples of tough movement in the training data to 18 examples (including one example with *that* as the

tough mover) by performing some additional hand annotation of PTB sentences. This bolstered the tough adjective supertag to 10th position, while the tough movement supertag for *that* now appeared in 28th position. This was not enough to enable the parser to correctly recover the tough movement analysis, but it does suggest that adding even more examples of the relevant type would ultimately enable it to do so.

In defence of the supertagger, it should be noted that tough movement is currently not found at all in the automatically generated portion of MGBank (because this analysis was introduced after the treebanking had already been carried out), only in the hand-crafted portion, meaning that this construction is drastically under-represented in the supertagger's training data at present.

Our A\* CCG parser scored 1/2 (the same as Clark et al. (2004)) on the tough movement examples; its higher performance is no doubt due to the much smaller tag set and hence the higher supertagging accuracy. CCG also does not require special supertags for tough-moved DPs, hence it avoids this particular data sparsity problem.

- ✓ 1. The survey found that nearly half of Hong Kong consumers espouse *what* it *identified* as materialistic values compared with about one-third in Japan and the U.S.
- ✓ 2. *What* she *did* was like taking the law into your own hands
- ✓✓ 3. We work damn hard at *what* we *do* for damn little pay and *what* she *did* cast unfair aspersions on all of us
- ✓ 4. There may be others doing *what* she *did*
- ✓ 5. The U.S. wants the removal of *what* it *perceives* as barriers to investment ; Japan denies there are real barriers
- ✓ 6. But they have n't clarified *what* those might *be*
- ✓ 7. Deregulation has effectively removed all restrictions on *what* banks can *pay* for deposits as well as opened up the field for new products such as high - rate CDs
- ✓ 8. Mr. Martin said they have n't yet decided *what* their next move would *be* but he did n't rule out the possibility of a consent solicitation aimed at replacing Georgia Gulf 's board
- ✓ 9. *What* matters is *what* advertisers are *paying* per page and in that department we are doing fine this fall said Mr. Spoon
- w.o. 10. *What* this *tells* us is that U.S. trade law is working he said
- t.o. 11. The paper accused him of being a leading proponent of peaceful evolution a catch phrase to describe *what* China believes *is* the policy of Western countries to seduce socialist nations into the capitalist sphere
- t.o. 12. Despite the harsh exchanges the U.S. and China still seem to be looking for a way to mend relations which have deteriorated into *what* Mr. Nixon *referred to* as the greatest crisis in Chinese - American relations since his initial visit to China num years ago
- ✓ 13. Judge Ramirez num said it is unjust for judges to make *what* they *do*.
- ✓ 14. Judges are not getting *what* they *deserve*
- t.o. 15. Composer Marc Marder a college friend of Mr. Lane 's who earns his living playing the double bass in classical music ensembles has prepared prepared an exciting eclectic score that tells you *what* the characters are *thinking* and *feeling* far more precisely than intertitles or even words would
- ✓ 16. We have and I 'm sure others have considered *what* our options *are* and we 've had conversations with people who in the future might prove to be interesting partners

Figure 6.6: The 16 sentences with free object relative clause dependencies in section 00 of the PTB. Each tick indicates a point awarded for the correct identification of the extraction site of the *wh* word; t.o. indicates that the parser timed out before returning a parse, and w.o. indicates that the parser correctly identified an object relative dependency but extracted the wrong object of a double object verb. Note that sentence 3 contains two free object relative clauses.

1. It 's the petulant complaint of an impudent *American whom* Sony *hosted* for a year while he was on a Luce Fellowship in Tokyo – to the regret of both parties
- ✓ 2. It said the *man whom* it did not *name* had been found to have the disease after hospital tests
- ✓ 3. Commonwealth Edison now faces an additional court-ordered *refund* on its summer/winter rate differential collections that the Illinois Appellate Court has *estimated* at \$ num million
- ✓ 4. But Rep. Marge Roukema -LRB- R. N.J -RRB- instead praised the House 's acceptance of a new youth training wage a *subminimum* that GOP administrations have *sought* for many years
5. Democratic Lt. Gov. Douglas Wilder opened his gubernatorial battle with Republican Marshall Coleman with an abortion *commercial* produced by Frank Greer that analysts of every political persuasion agree *was* a tour de force
- ✓ 6. Against a shot of Monticello superimposed on an American flag an announcer talks about the strong *tradition* of freedom and individual liberty that Virginians have *nurtured* for generations
7. Another was Nancy Yeargin who came to Greenville in num full of the *energy* and *ambitions* that reformers wanted to *reward*
8. Mostly she says she wanted to prevent the *damage* to self - esteem that her low - ability students would *suffer* from doing badly on the test
- ✓ 9. Mrs. Ward says that when the cheating was discovered she wanted to avoid the morale - damaging public *disclosure* that a trial would *bring*
- ✓ 10. Mr. Sherwood speculated that the *leeway* that Sea Containers *has* means that Temple would have to substantially increase their bid if they 're going to top us
11. A high - balance *customer* that banks *pine for* she did n't give much thought to the rates she was receiving nor to the fees she was paying
- ✓ 12. Interviews with analysts and business people in the U.S. suggest that Japanese capital may produce the economic *cooperation* that Southeast Asian politicians have *pursued* in fits and starts for decades
13. Interpublic Group said its television programming *operations* – *which* it *expanded* earlier this year – agreed to supply more than num hours of original programming across Europe in num
- ✓ 14. Interpublic is providing the programming in return for advertising *time* *which* it said *will* be valued at more than \$ num million in num and \$ num million in num
15. Mrs. Hills said many of the num *countries* that she *placed* under varying degrees of scrutiny have made genuine progress on this touchy issue
- ✓ 16. The Japanese companies bankroll many small U.S. companies with promising products or ideas frequently putting their money behind *projects* that commercial banks wo n't *touch*
17. In investing on the basis of future transactions a role often performed by merchant banks trading companies can cut through the *logjam* that small - company owners often *face* with their local commercial banks
- ✓✓ 18. He described the situation as an escrow *problem* a timing *issue* *which* he said *was* rapidly rectified with no losses to customers
19. In CAT sections where students ' knowledge of two - letter consonant sounds is tested the authors noted that Scoring High concentrated on the same *sounds* that the test *does* – to the exclusion of other *sounds* that fifth graders should *know*
- ✓✓ 20. The events of April through June damaged the *respect* and *confidence* which most Americans previously *had* for the leaders of China

Figure 6.7: The 20 sentences with non-free object relative clause dependencies in section 00 of the PTB. Note that following Clark et al. (2004), two points are awarded if the correctly relativized NP is a coordinate complex.



## **Part III**

# **Conclusion**





# **Chapter 7**

## **Conclusion**

This chapter summarises the contributions of the previous chapters, and suggests some possible areas for future research.

## 7.1 Summary of contributions

The primary thesis put forward here was that the MG formalism can be used to build deep and precise yet (relatively) efficient wide-coverage statistical parsing systems which are able to directly incorporate many of the proposals from the mainstream Minimalist literature. A further thesis was that constructing wide-coverage Minimalist grammars and subjecting them to computational testing can be beneficial for Minimalist theory itself, necessitating rigorous formalisation, revealing incompatibilities between analyses of different constructions, and exposing previously unnoticed areas of under- and overgeneration.

**Chapter 2** provided some background on Chomskyan Minimalism and Stabler's MG formalism, highlighting various similarities and differences between the two frameworks, and summarising some of the important formal results for MGs from the literature. The MG formalism and the wide-coverage grammar presented here clearly differ from MP on certain matters of detail (for example, many fine-grained selectional features were used to prevent the grammar from overgenerating<sup>1</sup> that are either not used at all in MP or only mentioned in passing<sup>2</sup>). Nevertheless, they share its most fundamentally defining properties: movement and null heads; they also include the spec-head feature checking operations and case-driven movements of (early) MP, as well as a strict version of the Shortest Move Constraint and, as presented here, many of the other locality constraints from mainstream TG. As standardly implemented, MGs also adopt many of MP's assumptions about phrase structure and clause/nominal structure, although there has been a shift within MG research towards viewing the derivation tree, rather than the phrase structure tree, as the primary syntactic data structure. Overall, MGs, including the wide-coverage MG presented here, are sufficiently close to MP to be reasonably regarded as a formalisation of the essential aspects of the latter (Müller, 2016, pages 165-166).

**Chapter 3** presented the Extended Directional Minimalist Grammars formalism constructed for this project, and also included detailed discussions of the analyses of

---

<sup>1</sup>Additional constraints and features to prevent overgeneration over and above those used in the TG literature were also found to be needed in Friedman (1971) and Gross (1978).

<sup>2</sup>See e.g. Pesetsky (1996) on I-selectional features.

many of the most interesting constructions in the MGbank grammar (lexical categories for many more of the constructions are provided in Appendix A). As well as synthesising many of the existing proposals in the MG literature in order to handle, e.g., adjunction, head movement, covert phrasal movement, rightward movement, across-the-board phrasal movement, and so on, this chapter made several novel contributions to the MG formalism, including rules for excorporation and across-the-board head movement, lexical head coordination, agreement and fine-grained subcategorisation. Throughout this chapter, the issue of minimising overgeneration was a primary concern (most existing wide-coverage parsers leave much more of this work to the probabilistic model, but this was not possible initially because there was no data to train on), and in addition to many positive examples of constructions generated by the grammar, there were many asterisked negative examples which the grammar correctly blocks (under our idealisation that grammatical deviance is, at least in many cases, binary); a number of areas in which the grammar currently over- or undergenerates were also noted.

We saw several instances where decisions in one area of the grammar had a knock on effect for other areas of the grammar. For example, section 4.7.3, showed that adopting the Movement Theory of Control of Hornstein (2001) together with the standard MG assumption that syntactic structure building features are strictly ordered, further necessitates the adoption of the swoop movement proposal for case/phi driven movement of Epstein and Seely (2006) in order to avoid undergenerating successive cyclic control movement structures such as *Pete wants to try to help*. There were also areas of overgeneration which arose within the grammar, many of which were detected when the Autobank parser returned unreasonable structures for certain sentences, highlighting the utility of rigorous computational testing in grammar development. For example, section 4.2.2 showed that (again owing to our assumption that control is A-movement and that syntactic features are ordered) it was necessary to allow theta- and case-driven movement to block one another (via DSMC) in order to avoid overgenerating certain unwanted co-indexations in ECM structures. At the same time, this assumption turned out to be too strong for certain gerund and reflexive binding structures, as we saw in sections 4.2.3 and 4.2.4 respectively, which made it necessary to make certain adjustments to the formalism and the grammar.

**Chapter 5** presented the Autobank system, which was used to construct the first ever wide-coverage MG, presented in the previous chapter and in Appendix A, as well as MGbank, the first wide-coverage corpus of MG derivation and phrase structure trees. Constructing a wide-coverage grammar and treebank for Minimalism is

particularly challenging, owing to the higher complexity of Minimalist syntax relative to other formalisms (for example, there are many more (head, covert, sideward, remnant) movement operations and null heads in MP trees than are included in the PTB). The creation of the grammar and the treebank were necessary steps towards constructing an efficient statistical MG parser with wide-coverage. The Autobank system itself will also be made publicly available, which will allow researchers to modify MGbank or even create their own wide-coverage MG and treebank from scratch. This is particularly appropriate for Minimalism given that, as is often emphasised in the MP literature, there is no single unified Minimalist Theory, only a Minimalist Program.

**Chapter 6** presented the first ever wide-coverage MG parser, which is also arguably the first ever wide-coverage Transformational Grammar parser. The results of this initial attempt are optimistic. First, the accuracy on recovering syntactic and semantic dependencies predicted by the Minimalist syntax is relatively high considering the higher complexity of the mechanisms involved relative to other formalisms. In comparison to CCG, a formalism with a much longer history of wide-coverage parsing, the results currently lag behind both in terms of coverage and on the recovery of both global and unbounded long-distance dependencies (the test set for the global dependency evaluation was also somewhat easier than is standardly the case owing to the fact that the sentences in MGbank are on average 16.9 words long vs 21.7 words in the original PTB). This gap will likely narrow in the future as the size and quality of MGbank improves and as better probabilistic models are developed enabling MG parsers to return analyses for a higher number of sentences.

Another important and optimistic result of this investigation is that Minimalist Grammar parsing is not as asymptotically slow as may have been expected given its worst case time complexity. Worst case complexity results have previously been raised as one of the prime criticisms of TG theories. The results presented here show that the combination of a good neural probabilistic model and A\* search makes Minimalist parsing far less costly than the worst case analyses might seem to suggest, with an expected time complexity for the parser and model presented here of  $O(n^3)$ . Absolute parsing speed is currently slow by modern parsing standards, but there is a great deal of room for improvement in terms of re-implementing the parser in a faster language with the optimisations to the chart suggested in Stanojević (forthcoming).

## 7.2 Future Work

This dissertation and the resources it has presented should hopefully open up several avenues for future research. In the short term, the priority will be to continue developing the MGbank grammar and corpus, both in terms of its quality and its coverage. In terms of the grammar, the chief aim will now be to revise some of the more unsatisfactory analyses and also to reduce the number of categories it contains by introducing rules which allow it to better generalise. For example, instead of listing separate passive entries for each type of verb, a passivisation rule could be used to knock out the +CASE licenser of a transitive verb and change its TRANS property to INTRANS allowing it to combine with the [intrans] little *v*. This will of course lead to issues with overgeneration which will need to be carefully considered (for example *he expected Jack to help* can be passivised to *Jack was expected to help*, but *he wanted Jack to help* cannot be passivised in this way: *\*Jack was wanted to help*). Morphology may also be introduced at some point, which would allow for many more lexical entries to be eliminated, e.g. separate entries for *look, looks, looked*. The fewer lexical entries there are, the easier the grammar will be to modify and extend, and (arguably) the more psycholinguistically plausible it becomes owing to its increased capacity to generalise.

As for the treebank, the intention is to swap out the log-linear C&C supertagger which was used during the automatic treebanking process for the neural supertagger that was developed by Miloš Stanojević and used for the wide-coverage A\* parser, and which has already been modified by Miloš so that it can take the CCG supertags as auxiliary input tags during the treebanking process. This neural supertagger should have much higher accuracy than the older C&C supertagger, which should result in greater coverage and also more accurate candidate trees being proposed. Once MGbank has been regenerated using this supertagger and the CKY MG parser, as described in chapter 5, the A\* parser can then be retrained and used to further extend the treebank, particularly with respect to the longer sentences of the PTB, many of which are currently not included in MGbank owing to the fact that the exhaustive CKY MG parser tended to time out on these (though using the complex supertags rather than the atomic MG lexical categories during CKY parsing did enable the parser to generate analyses for some sentences up to length 50).

As well as improving the data, there is considerable scope for developing improved probabilistic models for wide-coverage MG parsing. One issue which was identified in chapter 6 was that the method for anchoring null heads to overt heads that was used

leads to too much data sparsity. This will be ameliorated to some extent by the further development of MGbank, but doubling or even tripling the data will not be enough to overcome the problem entirely. The parser's performance on specific constructions could be improved by adding more examples of that specific construction to the data, but a possible downside to this is that it would result in unnatural distributions which may negatively impact the parser's performance in unforeseen ways. One potential solution, which was suggested in chapter 6, would be to allow the null heads to enter the agenda as separate atomic items, and to have the supertagger assign them a probability based on the likelihood of them appearing in the derivation given the sentence.

Aside from the data sparsity issue, there is the fact that the simple model used here currently underdetermines the derivation with respect to some adjunction and coordination attachment ambiguities. For example, attaching a temporal adjunct to an embedded TP may result in the same expression as attaching it to the matrix clause TP would, and as both expressions will have the same score (because the same supertags are involved in either case), the model will be unable to distinguish between these two options and so at present one is simply dropped arbitrarily. For CCG, Lewis and Steedman (2014b) suggest augmenting the simple supertag factored model with a head-dependency model for breaking such ties, and this could equally be done for MG, though it would of course increase the worst case complexity of the parser, just as adding a head dependency model increases the worst case complexity of CKY CFG parsing from  $O(n^3)$  to  $O(n^5)$  (owing to the necessity of keeping track of the span position of the head word for each constituent).

As an alternative to A\* search, it would be interesting to implement the proposal for a transition-based MG parser made in Stanojević (2016). For generating a single parse, this parser has a worst case complexity of just  $O(n^2)$  and a best case complexity of  $O(n)$ , which it achieves by approximately searching through the search space using a probabilistic beam search. The approximate nature of the search, coupled with the strength of the grammar, do mean that there is no guarantee that this parser will find a parse, even if there is one to be found (unlike in the case of dependency grammars which lack the hard constraints of a formal grammar entirely and so will never hit a dead end during parsing) and so its success will very much depend on the quality of the probabilistic search. The effort may be worthwhile, however, because as Stanojević (2016) notes, this MG parser would be, in theory at least, as asymptotically efficient as parsers based on much less expressive formalisms, such as dependency and CFG parsers. Stanojević (2016) concludes by stating that 'in order for this transition to

become a reality, a necessary next step is the creation of a scoring model, as well as the creation of a Minimalist treebank on which the scoring model will be trained.’ The MGbank corpus presented in this dissertation makes training a scoring model for this parser viable for the first time. Of course, it could also potentially be used to train a neural end-to-end parser of the sort presented in Vinyals et al. (2015).

It is hoped that the new resources presented in this dissertation will also prove useful to the sorts of psycholinguistic investigations mentioned in the introduction. The bottom-up and non-incremental A\* parser and probabilistic model used for the current project were chosen because of their simplicity and efficiency, but they are clearly not plausible from a psycholinguistic perspective. There have recently been a number of interesting proposals in the MG literature for more psycholinguistically plausible, top-down (Stabler, 2013b) and left-corner (Hunter, 2018; Stanojević and Stabler, 2018) incremental parsers. The parsers in Stabler (2013b) and Stanojević and Stabler (2018) make explicit reference to a probabilistic beam search enabling the parser to pursue the most likely analyses at each step, and this beam can now be properly implemented using MGbank as training data.

The probabilistic model used for the A\* parser presented here was discriminative, assigning probabilities to supertags given the sentence. However, for psycholinguistically-oriented parsers, a generative model is usually preferred, as these parsers are generally evaluated according to how well their surprisal for each incoming word in the sentence correlates with empirical facts about processing difficulty with respect to different constructions (Hale, 2003). Hunter and Dyer (2013) argue that the generative PMCFG model proposed in Hale (2003), while simple, leads to over-parameterisation owing to the fact that it treats MG derivation tree non-terminals as MCFG atomic categories. As a result, the system necessarily models different instantiations of the same MG operation independently, so that, for example, merging an adverb with a VP containing a *wh* mover will have a different probability from merging the same adverb with a VP that is similar to the first except that it does not contain the *wh* mover. As an alternative, Hunter and Dyer propose a locally normalised log-linear generative model that is more faithful to the derivational processes of the MG formalism. This model could potentially be integrated with the aforementioned incremental parsing algorithms and trained on MGbank.

So far the parser presented here has only been evaluated on a very limited set of construction types. It would be interesting for future research to perform additional evaluations, for example on the entire unbounded dependency test corpus (which in-



cludes right node raising coordination) of Rimell et al. (2009). However, as the results on tough movement demonstrated, there needs to be a sufficient number of examples in the training data for these more complex constructions in order for the system to stand a chance of recovering them, and this is not currently the case for many constructions (including right node raising) owing to the fact that MGbank was itself created using a parser which performed better on some constructions than others. Sections 5.6-5.6.3.13 on the evaluation of MGbank also revealed that some constructions in the corpus currently have fairly low precision and/or recall. Further development of MGbank is therefore required in order to increase the numbers and quality of these types of constructions in the automatically generated portion of the training data before more comprehensive evaluations would be worthwhile. At that point it would be useful to construct a test suite of constructions on which to evaluate MG parsers, and a good starting point for this could be to convert the test suite of the HPSG *incr tsdb* project (Oepen, 2001) into the MGbank formalism. It would also be interesting to evaluate the MG parser on the grammaticality judgement task using the Corpus of Linguistic Acceptability (CoLA; Warstadt et al. 2018).

Finally, as noted in the introduction, there have been some preliminary investigations into automatic grammar induction with MGs (Stabler, 1998; Bonato and Retoré, 2001; Portelance et al., 2017). The wide-coverage MG presented here should open up new avenues of research in this domain by enabling simulations with more realistic final state target grammars, while MGbank will enable more accurate modelling of the sorts of structural regularities and noise to which human learners are apparently sensitive.

**Part IV**  
**Appendices**



# Appendix A

## A selection of constructions and their lexical entries in MGBank

### A.1 Introduction

This appendix lists examples of lexical entries for many (though by no means all) of the constructions in MGBank, in order to give a broader flavour of the grammar than was possible in the main text. Note that all fine-grained selectional and agreement features are included here, in contrast to in the main text where categories were simplified to make the exposition and trees simpler. Note that there are separate entries in the MGBank category inventory for each morphological form of verbs, nouns, determiners and pronouns, marked with properties like +3SG, -3SG, 3PL, NOM, ACC etc, but only one or two members of each paradigm will be listed below. The same is true for the various null heads, such as [adjunctizer] and [topicalizer] heads etc, as well as for the many different types of coordinator category; in all cases, only a handful of the relevant categories are included below. A fuller and discursive description of the grammar is not feasible here, but all the lexical categories contained in MGBank, along with the derivation trees and phrase structure trees they generate, can be viewed using Autobank's search facilities (both the comments associated with each category, as well as its features, can be searched for).

It is important to bear in mind that this grammar is currently still a work in progress, and probably will be for some years to come. A few of the analyses (e.g. possessives, imperatives and the copula) are currently undergoing revision, and this is indicated below. The reader will note that there are often a great many fine-grained selectional properties and requirements used with a given lexical entry. Unfortunately, it is not

feasible to explain the motivation for all of these features here; they are not intended to constitute any kind of serious theoretical claim about the human grammar, and many of them were added in a somewhat ad hoc manner in order to reduce overgeneration and make parsing more efficient during the treebanking process, where no (good) statistical model was available for constraining the search space. As such, many of these features can be viewed as proxies for a statistical model, and in fact chapter 6 showed that it was possible to remove almost all of these at test time with only a slight reduction in the precision and F1 of the parser. Of course, the complex supertag categories which anchor null heads to overt ones essentially precompiled many of the constraining effects of the fine-grained selectional restrictions into the complex supertags, but a good model of the derivation itself should have a similar constraining effect for atomic category MG parsing. Autobank includes a view of derivation trees in which all fine-grained features have been stripped out, leaving just the structure-building features.

## A.2 Basic n-place predicate constructions

### A.2.1 little v heads

[trans] :: >v{+[STRANS|TRANS].[+DISJ|-NEG].x}= +self! +wh{+OVERT.~NOM}?  
=D{-EXPL.[+OVERT|-OP].x} lv{NULL.TRANS.x}

[intrans] :: >v{+INTRANS.x}= +LOC! lv{INTRANS.NULL.x}

### A.2.2 Unergative intransitive verbs

laughs :: v{+3SG.OVERT.PRES.TRANS}

laughed :: v{OVERT.PAST.PERF.TRANS}

eaten :: v{OVERT.PERF.TRANS}

### A.2.3 Unaccusative verbs

arrived :: d{-EXPL.[+OVERT|-OP]}= v{INTRANS.OVERT.PAST.PERF}

falling :: d{-EXPL.[+OVERT|-OP]}= +num{+NONE}? v{INTRANS.OVERT.PROG}

total :: d{+OVERT}= +CASE{+ACC.+OVERT} =d{+OVERT.-EXPL}  
 v{-3SG.INTRANS.OVERT.PRES} % unaccusative verb that takes a second internal  
 DP argument (*they total 3 million barrels*).

### A.2.4 Transitives

helped :: d{+OVERT.-EXPL}= +CASE{+ACC.+OVERT} v{OVERT.PAST.PERF.TRANS}

took :: d{+OVERT.-EXPL}= +CASE{+ACC.+OVERT} v{OVERT.PAST.TRANS}

helped :: d{-EXPL.[+OVERT|-OP]}= +num{+NONE}? v{INTRANS.OVERT.PASS}  
 % Passivised transitive.

### A.2.5 Ditransitives

gives :: d{+OVERT.-EXPL}= +CASE{+ACC.+OVERT} =d{+OVERT}  
 +CASE{+ACC.+OVERT}  
 v{-3SG.DO.U.OVERT.PRES.TRANS}

given :: d{+OVERT.-EXPL}= +CASE{+ACC.+OVERT} =d{+OVERT}  
 +CASE{+ACC.+OVERT} v{DO.U.OVERT.PERF.TRANS}

given :: d{-EXPL.[+OVERT|-OP]}= +CASE{+ACC.+OVERT} =d{-EXPL.[+OVERT|-  
 OP]} +num{+NONE}? v{INTRANS.OVERT.PASS} % passivized ditransitive

### A.2.6 Prepositional datives

gives :: p{+VCOMP.-DAT.-PASS}= =d{-EXPL.[+OVERT|-OP]} +CASE{+ACC.+OVERT}  
 v{-3SG.OVERT.PRES.TRANS}

given :: p{+OVERT.+VCOMP}= +self! =d{-EXPL.[+OVERT|-OP]} +num{+NONE}?  
 v{INTRANS.OVERT.PASS} % passivized prepositional dative

### A.2.7 Reporting verbs

said :: c{+MAIN.-SUB}= v{OVERT.PAST.PERF.REPORT.STRANS} % direct re-  
 ported speech verb

said :: c{+DECL.+FIN.+SUBORD.-SUB}=v{IREPORT.OVERT.PAST.PERF.STRANS}  
 % indirect reported speech verb

### A.2.8 4-place verbs

bet :: c{+DECL}=d{+OVERT.-EXPL} +CASE{+ACC.+OVERT} =d{+OVERT}  
 +CASE{+ACC.+OVERT} v{OVERT.PAST.PERF.TRANS} % *Jack bet Mary 5 pounds  
 that he would win.*

### A.2.9 The copula

The copula is one of the constructions currently undergoing revision. The lexical categories shown here are for the new analysis, which involves a prdP projection (Mikkelsen, 2005).

is :: prd{+OVERT}=+self! +LOC! lv{+3SG.AUX.BE.INTRANS.LV.OVERT.PRES}

[predicatizer] :: adj{+OVERT.+PRD.x}=d{-EXPL.[+OVERT|-OP]} prd{x}

[predicatizer] :: p{+OVERT.+PRD.-DAT.x}=d{+OVERT.-EXPL} prd{x} % *Jack is  
 in the garden, There seem to be fairies in the garden.*

[predicatizer] :: d{+OVERT.-DECL.-EXPL.-INF.x}=+case{+OVERT} =d{-EXPL.[+OVERT|-  
 OP]} prd{x}

[predicatizer] :: d{+OVERT.-DECL.-EXPL.-INF.x}= prd{x} % This is used where  
 subject is expletive *there* and no following PP after copula *There are several problems.*

[predicatizer] :: c{+[IMP|SUBORD]}=d{-EXPL.[+OVERT|-OP]} prd{x} % *One  
 big obstacle is that few drugstores develop the film anymore.*

[predicatizer] :: lv{+OVERT.+[PASS|PROG].x}= prd{x} % Used where passive is  
 coordinated with an adjective (*the causes of homelessness are complex and poorly  
 understood*).

and :: prd{+OVERT}=prd{+OVERT.x} prd{COORD.x} % ‘Unlike’ constituent co-  
 ordinator following copula (*Jack is happy and in the garden*).

### A.2.10 'Restructuring' verbs taking direct vP or VP complement

see :: lv{+OVERT.+PROG}= +CASE{+ACC.+OVERT} v{-3SG.OVERT.PRES.TRANS}  
 % *We see protestors marching around.*

having :: lv{+OVERT.+PASS}= +CASE{+ACC} v{OVERT.PROG.TRANS} % *Firms face the prospect of having their credit ratings lowered.*

seen :: lv{+BARE.+OVERT}= +CASE{+ACC} v{OVERT.PERF.TRANS} % *I have seen one or two men die.*

seen :: lv{+OVERT.+PROG}= v{INTRANS.OVERT.PASS} % (*He was later shown on television fielding questions.*) % Passive verb taking progressive VP.

continued :: lv{+OVERT.+PROG}= =d{-EXPL.[+OVERT|-OP]}  
 v{INTRANS.OVERT.PAST.PERF} % unaccusative verb taking progressive vP complement (*it continued getting stronger*).

## A.3 Passives and expletive it

### A.3.1 Passive auxiliaries

were :: lv{+PASS}= voice{+[1PL|2PL|2SG|3PL].AUX.BE.OVERT.PAST}

being :: lv{+PASS}= +num{+NONE}? +EPP{+ASSOC}! voice{AUX.OVERT.PROG}

[pres] :: >voice{+PRES.-FOC.[+AUX|-ELLIP].x}=  
 +CASE{+NOM.+OVERT.-EXTRAP.x} t{FIN.IND.NULL.PRES.x} % T head selecting for voiceP.

### A.3.2 Passivised transitive verb

beaten :: d{-EXPL.[+OVERT|-OP]}= +num{+NONE}? v{INTRANS.OVERT.PASS}

### A.3.3 Expletive Passive/Progressive

[exp] :: d{+ASSOC.+OVERT.-EXP.x}= +num{y} d{EXP.x} -epp{ASSOC} -num{y}  
 % Adds -epp licensee to DP associate to move it to left of passive or unaccusative pro-



gressive verb (*There were several monuments constructed, There were several people arriving*).

[epp] :: >lv{+INTRANS.+OVERT.+[PASS|PROG].-EPP.x}= +EPP{+ASSOC} lv{EPP.x}  
 %Attracts the DP associate to a position left of the passive or unaccusative progressive verb.

Note that expletive *there* with non-unaccusative progressive verbs is simpler to generate, as here the DP associate simply remains in its base generated spec-vP position (*There were several people eating their dinner*). The issue with unaccusatives is that the base-generated position is post-verbal.

### A.3.4 Impersonal passives

it :: d{3SG.EXPL.IT.OVERT} -case{3SG.ACC.NOM.OVERT} % When expletive *it* appears as the subject of a verb which naturally takes a CP complement, it is generated in spec-CP using the associator below (analysis adapted from Stroik (1996)) *It was hoped that other Japanese would then follow, it seemed that Number 10 was refusing to comment.*

[associator] :: c{+OVERT.-IT.x}= =d{+EXPL.+IT.-EXTRAP} c{IT.x}

### A.3.5 *it*-extraposition

it :: c{+EXTRAP.+SUBORD.x}= d{3SG.IT.OVERT.TEST.x} -case{3SG.ACC.NOM.OVERT}  
 % Here, expletive *it* is treated as a determiner that takes a rightward moving CP as its complement.

[extraposer] :: c{-EXTRAP.-FOC.-SHELL.-TOPIC.x}= c{EXTRAP.NOMOD.NULL.x}  
 t{EDGE} % *It* in subject position (*It doesn't take much to get burned, it was obvious that he was lying.*)

[extraposer] :: c{+SUBORD.-EXTRAP.-FOC.-NOMOD.-SHELL.-TOPIC.x}=  
 c{EXTRAP.NOMOD.NULL.x} v{EDGE} % *It* in object position (*The companies will leave it up to the marketplace to decide.*)

See also section A.9.2 on clefts.

### A.3.6 Pseudo Passive

unheard :: p{+OVERT.+PASS}<= =d{-EXPL.-EXTRAP.[+OVERT|-OP]}  
 v{INTRANS.OVERT.PASS} % *abrupt departures aren't unheard of, The hat was sat on.*

of :: p{OVERT.PASS} % Special preposition for pseudo passive which does not select object or check case, but gets incorporated to the passive verb that selects it.

## A.4 Particle verbs

Treating some particle verbs as involving incorporation of the particle is discussed in Radford (1988) and suggested to be a case of part-to-V head movement in the context of MGs in Stabler (2001b).

looked :: part<= =d{+OVERT.-PRO} +CASE{+ACC.+OVERT.-PRO}  
 v{OVERT.PAST.PERF.TRANS} % Particle verb incorporating the particle, so no pre-modifier of the particle is allowed (*legal authorities cranked (\*right) up the investigation*); pronominal object also not allowed (*\*legal authorities cranked up it/that*)

up :: part{OVERT}

lashed :: part{-DIS}= v{OVERT.PAST.PERF.TRANS} % Particle verb not incorporating the particle and therefore allowing pre-modifier for the particle (*he lashed (right) out*).

looked :: part{-DIS}= =d{+OVERT.-EXPL} +CASE{+ACC.+OVERT}  
 v{OVERT.PAST.PERF.TRANS} % Particle verb not incorporating the particle and taking DP object, which appears between the verb and the particle, with possible pre-modifier for the particle (*It might have scared us (right) off*).

wound :: part<= =d{-EXPL.[+OVERT|-OP]} =lv{+OVERT.+PROG}  
 v{INTRANS.OVERT.PAST.PERF} % Unaccusative 'restructuring' particle verb selecting progressive vP and incorporating particle (*He wound (\*right) up helping*).

go :: part{-DIS}= =d{-EXPL.[+OVERT|-OP]}  
 v{BARE.INTRANS.OVERT} % Unaccusative particle verb; particle is not incorporated so pre-modifier is possible (*go (right) back*).

## A.5 The inflection domain

### A.5.1 Progressive auxiliaries

is :: lv{+PROG.x}= +LOC! prog{+3SG.AUX.BE.OVERT.PRES.x}

is :: voice{+PROG}= +LOC! prog{+3SG.AUX.BE.OVERT.PRES} % *The effort is being led by Contel.*

[pres] :: >prog{+PRES.[+AUX|-ELLIP].x}= +CASE{+NOM.+OVERT.-EXTRAP.x} t{FIN.IND.NULL.PRES.x} % Version of T head selecting progP.

### A.5.2 Perfective auxiliaries

has :: lv{+PERF.x. PAST}= perf{+3SG.AUX.OVERT.PRES.x}

has :: voice{+PERF.-FOC}= perf{+3SG.AUX.OVERT.PRES} % *No price for the new shares has been set.*

has :: prog{+PERF}= perf{+3SG.AUX.OVERT.PRES} % *Supply has been increasing, Supply has been being increased.*

[past] :: >perf{+PAST.[+AUX|-ELLIP].x}= +CASE{+NOM.+OVERT.-EXTRAP.x} t{FIN.IND.NULL.PAST.x} % Past T head selecting perfP complement.

### A.5.3 Negation

[neg] :: >t{x}= =negs{+OVERT} +pol! neg{INV.UPPER.x} % Upper neg head (*mightn't he not have done that?*)

[neg] :: >perf{x}= =negs{y} +pol! neg{MID.NULL.x.y} % Standard neg head (*he didn't do that*). Licenses polarity items (*he didn't do anything*).

not :: negs{NOT.OVERT} % The overt negative particle, appears in spec-negP.

never :: negs{TMP} % *never* is also currently treated as appearing in spec negP and hence licensing a polarity item (as the neg head has the +pol! feature but requires an overt specifier).

[pres] :: >neg{+MID.+PRES.[+AUX|-ELLIP].[+AUX|-NOT].[+COORD|-INERT].x}=

+CASE{+NOM.+OVERT.-EXTRAP.x} t{FIN.IND.NULL.PRES.x} % Present T head selecting for negP.

does :: neg{+BARE.+MID.+NOT.[+COORD|-INERT].x}=  
+CASE{+3SG.+NOM.+OVERT.x} t{AUX.CASE.DO.FIN.IND.OVERT.PRES.x} %  
Overt present T head selecting negP.

#### A.5.4 Modal verbs

can :: lv{+BARE.x}= +pol! mod{+NOM.AUX.MD.OVERT.x}

will :: voice{+BARE.x}= +pol! mod{+NOM.AUX.MD.OVERT.x} %*Almost all remaining uses of asbestos will be outlawed.*

will :: prog{+BARE.x}= +pol! mod{+NOM.AUX.MD.OVERT.x} %*They will be going for a full bid.*

might :: perf{+BARE.x}= +pol! mod{+NOM.AUX.MD.OVERT.x} %*It might have scared us off.*

can :: neg{+BARE.+MID.x}= +pol! mod{+NOM.AUX.MD.OVERT.x} %*You can't hold back technology*

[pres] :: >mod{[+AUX|-ELLIP].x}= +CASE{-EXTRAP.x} +pol!  
t{FIN.IND.NULL.PRES.x} % Present T head selecting modP

#### A.5.5 Subjunctive Mood

[sub] :: >lv{+BARE.+BE.-ELLIP.x}= mod{+NOM.FIN.NULL.SUB.x} (*I demand that he be there on time*)

[sub] :: lv{+BARE.-BE.-ELLIP.x}= mod{+NOM.FIN.NULL.SUB.x} % (*I demand that he go there tonight.*) These two [sub] heads should probably be merged into one because *be* does not appear to undergo movement to T in subjunctives given adverb placement: *I demand that he (definitely) be (\*definitely) there on time.* cf *he was definitely there on time.*

### A.5.6 Tense

[pres] :: >lv{+BE.+PRES.[+AUX|-ELLIP].x}= +CASE{+NOM.+OVERT.-EXTRAP.x}  
t{FIN.IND.NULL.PRES.x} % Present tense null T head selecting BE vP and triggering v-to-T head movement.

[pres] :: >perf+PRES.[+AUX|-ELLIP].x= +CASE{+NOM.+OVERT.-EXTRAP.x}  
t{FIN.IND.NULL.PRES.x} % Present tense null T head selecting perfP and triggering perf-to-T head movement.

[pres] :: lv{+PRES.-BE.[+AUX|-ELLIP].x}= +CASE{+NOM.+OVERT.-EXTRAP.x}  
t{FIN.IND.NULL.PRES.x} % Present tense null T head selecting non-BE vP with no head movement.

[pres] :: lv{+PRES.-BE.[+AUX|-ELLIP].x}= +pers{+OVERT.x} +num{+OVERT.x}  
+EPP{+NOM.+OVERT} t{FIN.IND.NULL.PRES.x} % Present tense null T head selecting non-BE vP with no head movement and attracting expletive *there* subject.

[past] :: lv{+PAST.-BE.-DO.[+AUX|-ELLIP].x}= +CASE{+NOM.+OVERT.-EXTRAP.x}  
t{FIN.IND.NULL.PAST.x} % Past tense null T head selecting non-BE vP with no head movement.

does :: lv{+BARE.-BE.x}= +CASE{+3SG.+NOM.+OVERT.x}  
t{AUX.CASE.DO.FIN.IND.OVERT.PRES.x} % Present tense overt T head selecting non-BE vP complement.

did :: lv{+BARE.-BE.-IMP.x}= +CASE{+NOM.+OVERT.x}  
t{AUX.CASE.DO.FIN.IND.OVERT.PAST.x} % Past tense overt T selecting non-BE vP.

does :: lv{+BARE.-BE.x}= +pers{+OVERT.x} +num{+OVERT.x} +EPP{+NOM.+OVERT}  
t{AUX.CASE.DO.FIN.IND.OVERT.PRES.x} % Present tense overt T head selecting non-BE vP and attracting expletive *there* subject.

### A.5.7 Emphatic *do*

do :: lv{+BARE.x}= +CASE{+3SG.+NOM.+OVERT.x} t{AUX.DO.EMPH.IND.OVERT.PRES}

[int] :: >t{+FIN.-IMP.-INV.[+EMPH|-DO].x}= +WH{+NOM.+OVERT.-ECHO.-REL.y}

+negs{+NONE}? +foc{+NONE}? +TOP{+IF}! +pol! c{INT.MAIN.NULL.x.y} %  
 Interrogative C head for subject wh-questions, disallows *do*-support but allows em-  
 phatic *do* (*Who **does** go there?*)

## A.6 More A-movement constructions

### A.6.1 subject raising verbs

seems :: t{+IND.+INF.-RREL}= v{+3SG.INTRANS.OVERT.PRES}

seems :: t{+IND.+INF.-RREL}= =p{+TO} v{+3SG.INTRANS.OVERT.PRES} % Used  
 when the raising verb takes a PP complement in addition to the TP complement (*He  
 seems to me to like her*).

### A.6.2 Raising adjectives

likely :: t{+INF.-RREL}= adj{OVERT.PRD}

### A.6.3 Exceptional Case Marking (ECM) (object raising)

expected :: t{+IND.+INF.-RREL}= +CASE{+ACC.+OVERT}!  
 v{OVERT.PAST.PERF.TRANS}

expected :: t{+IND.+INF.-RREL}= +num{+OVERT.x} +pers{+OVERT.x}  
 +EPP{+ACC.+OVERT} v{OVERT.PAST.PERF.TRANS} %ECM verb when exple-  
 tive *there* is the ECM object (*You can expect there to be consequences*).

expected :: t{+INF.-RREL}= v{INTRANS.OVERT.PASS} % Passivised ECM verb.

### A.6.4 *want*-type subject control

want :: c{+DECL.+IND.+INF.-OP}= =p{+DAT} v{BARE.OVERT.TRANS}

### A.6.5 *promise*-type subject control

promises :: c{+DECL.+IND.+INF.-OP}= =p{+DAT} v{+3SG.OVERT.PRES.TRANS}

[dat] :: d{+OVERT.-DECL.-EXPL.-INF.-PHI.-PROG}= +case{x}

p{DAT.NOMOD.NULL.x} % Absorbs the D and -case features of the DP object of *promise* and prevents (D)SMC violation.

### A.6.6 Object control

persuade :: c{+DECL.+IND.+INF.-OP}= +self! =D{+OVERT} +CASE{+ACC.+OVERT} v{-3SG.OVERT.PRES.TRANS}

persuaded :: c{+DECL.+IND.+INF.-OP}= =D{+OVERT} v{INTRANS.OVERT.PASS}  
% Passivised object control verb.

### A.6.7 Adjunct control

[adjunctizer] :: c{+INF.+OVERT.-ELLIP.-FOR.-PASS}= +log! ≈lv{-NOMOD.-PASS}  
% Right adjoins INF purposive clause to main clause (*That sector is stepping forward to pick up the slack*).

[adjunctizer] :: c{+INF.+OVERT.+PASS.-ELLIP.-FOR.-NOMOD}= +log! ≈lv{+PASS.-NOMOD} % Right adjoins INF passive clause to passive main clause with adjunct control (*he was taken away from his cell to be questioned*).

[topicalizer] :: c{+OVERT.+[GER|INF|VMOD].-ELLIP.-EXTRAP.-FOR}= +log! ≈lv{-NOMOD.-PASS} -top % Right adjoins INF clause to main clause with DP ATB moving out of the INF adjunct clause, then the INF clause undergoes remnant topicalization movement (*When referred to the questions that matched, he said it was coincidental*).

### A.6.8 Non-obligatory control

[pro-d] :: D{1PL.1SG.2PL.2SG.3PL.3SG.ACC.MASC.NOM.NOMOD.NULL.PR.D.PRO}  
% NOC PRO

### A.6.9 for-to infinitivals

for :: t{+INF.-RREL.x}= +CASE{+ACC.-FOC.-TOP.[+ECHO|-WH]} p{FOR.x} %  
P part of prepositional complementizer *for*.

for :: t{+INF.-RREL.x}= +num{+OVERT.x} +pers{+OVERT.x} +EPP{+ACC.+OVERT}  
p{FOR.x} % P part of prepositional complementizer *for* used when the object of *for*

is expletive *there* (*I expected for there to be consequences*).

[decl] :: >p{+FOR.-GER.x}= +negs{+FOC}? c{DECL.OVERT.SUBORD.x} % C part of prepositional complementizer *for*; triggers head movement of *for* to place it to the left of its object.

### A.6.10 Floating quantifiers

all :: q{FLOAT.NOMOD.PL.SG} -epp{EDGE.FLOAT} -num{OVERT.PL.SG} % *It all adds up to a cold winter here.*

both/each :: q{FLOAT.NOMOD.PL} -epp{EDGE.FLOAT} -num{OVERT.PL} % *The men must both have loved her.*

[float] :: d{+3SG.+DEF.+OVERT.-EXPL.-EXTRAP.-FL.-NAME.-OP.-Q.x}= +case{+OVERT.-FL.y} =q{+FLOAT.+SG} D{FL.x} -pers{3.FL.OVERT} -epp{y} % Associates a singular floating quantifier with a singular DP antecedent.

[float] :: d{+3PL.+DEF.+OVERT.-EXPL.-EXTRAP.-FL.-OP.-Q.x}= +case{+OVERT.-FL.y} =q{+FLOAT.+PL} D{FL.x} -pers{3.FL.OVERT} -epp{y} % Associates a plural floating quantifier with a plural DP antecedent.

[epp] :: >perf{+OVERT.-EPP.x}= +EPP{+FLOAT} perf{EPP.x} % Used to attract the floating quantifier to spec-perfP (*The men must both have loved her.*)

[epp] :: >prog{+OVERT.-EPP.x}= +EPP{+FLOAT} prog{EPP.x} % Used to attract the floating quantifier to spec-progP (*It will all be adding up to a cold winter.*)

[epp] :: >voice{+OVERT.-EPP.x}= +EPP{+FLOAT} voice{EPP.x} % Used to attract the floating quantifier to spec-voiceP (*It will all be added up at the end of the winter.*)

### A.6.11 Locative Inversion

[-sbj] :: d{+3PL.+OVERT.-EXPL.-EXTRAP.-GER.-PHI.-REL.x}= +case{+OVERT.y} d{PHI.x} -pers{3.y} -num{ACC.NOM.PL.y} % Converts case feature of plural DP to -pers -num sequence so that it does not raise overtly to spec-TP subject position.

[-sbj] :: d{+3SG.+OVERT.-EXPL.-EXTRAP.-GER.-PHI.-REL.x}= +case{+OVERT.y} d{PHI.x} -pers{3.y} -num{ACC.NOM.SG.y} % Converts case feature of singular DP to -pers -num sequence so that it does not raise overtly to spec-TP subject position.



[sbj] :: p{+LOC.+OVERT.-DAT.-FOC.x}=p{EPP.NOMOD.x} -loc -epp{NOM.OVERT}  
 % Adds -loc -epp sequence to a locative PP, allowing it to raise to subject position in  
 locative inversion (*Behind all the hoopla is some pretty heavy-duty competition*)

### A.6.12 Expletive *there*

there :: d{3.EXPL.NOMOD.OVERT.PRO.THERE} -loc{EDGE} -pers{3.OVERT}  
 -epp{ACC.NOM.OVERT}

[associator] :: d{+3SG.+INDEF.+OVERT.-ASSOC.-WH.x}= +case{+OVERT.y}  
 +negs{+NONE}? =d{+3.+EXPL.+[LOC|THERE].-PART} D{ASSOC.NOMOD.x}  
 -num{SG.y} % Associates *there* with a singular indefinite DP.

[associator] :: d{+3PL.+INDEF.+OVERT.-ASSOC.-WH.x}= +case{+OVERT.y} +negs{z}  
 =d{+3.+EXPL.+[LOC|THERE].-PART} D{ASSOC.NOMOD.x} -num{PL.y} -negs{z}  
 % Associates *there* with a plural indefinite DP.

[associator] :: d{+3SG.+INDEF.+OVERT.+WH.-ASSOC.x}= +case{+OVERT.y} +wh{z}  
 =d{+3.+EXPL.+[LOC|THERE].-PART} D{ASSOC.NOMOD.x} -num{SG.y} -wh{z}  
 % Used when the singular DP associate is a wh item (*What sense is there in that?*).

[associator] :: d{+3PL.+INDEF.+OVERT.+WH.-ASSOC.x}= +case{+OVERT.y} +wh{z}  
 =d{+3.+EXPL.+[LOC|THERE].-PART} D{ASSOC.NOMOD.x} -num{PL.y} -wh{z}  
 % Used when the plural DP associate is a wh item (*What other options are there?*).

### A.6.13 Clausal subjects

[det] :: c{+SUBORD.+[INF|THAT].-EXTRAP.-REL.-RELAT.-TOPIC.x}=  
 +wh{+NONE}! +N{+NONE}! +top{+NONE}! +foc{+NONE}! +case{+NONE}!  
 +pol{+NONE}! d{3SG.DEF.NOM.NOMOD.OVERT.PR.D.x}  
 -case{3SG.DEF.NOM.OVERT.x} % Transforms CP into DP so it can appear in nom-  
 inative case subject position (*That he did that surprises me, For him to do that would  
 be unfair*).

### A.6.14 Reflexive and Reciprocal anaphors

myself :: self{+1SG}= D{1SG.ANA.OVERT} -case{1SG.ACC.ANA.OVERT}

himself :: self{+3SG.+MASC}= D{3SG.ANA.MASC.OVERT}  
-case{3SG.ACC.ANA.MASC.OVERT}

themselves :: self{+3PL}= d{3PL.ANA.OVERT} -case{3PL.ACC.ANA.OVERT}

[self] :: d{+[IMP|OVERT].-DECL.-EXPL.-INF.-PROG.-SELF.x}=  
+case{+[IMP|OVERT].y} self{x} -self{EDGE} D{SELF.x} -case{y} %

[self] :: d{+OVERT.-EXPL.-PROG.x}= self{x} -self{EDGE} % [self] head for where  
reflexive is inside PP complement and bound by an object, not subject.

[self] :: d{-EXPL.-IMP.-OVERT.-PROG.-SELF.x}= self{x} -self{EDGE} D{SELF.x}  
% Version of [self] head for where caseless NOC PRO is antecedent (*To love oneself  
is good*).

each :: d{+OTHER.-EACH.x}= +case{y} d{EACH.x} -case{y} % The *each* part of  
*each other* reciprocal.

other :: self{+3PL}= d{3PL.ANA.GEN.OTHER.OVERT}  
-case{3PL.ACC.ANA.GEN.OVERT} % The *other* part of *each other* reciprocal.

## A.7 Sentential force

### A.7.1 Declaratives

[decl] :: t{+FIN.-IMP.-INV.-MOD.-RREL.[+ELLIP|+EMPH|+NOT|-DO].x}=  
+foc{+NONE}? +wh{+NONE}? +negs{+NONE}? c{DECL.FIN.MAIN.NULL.x}  
% Main clause null declarative C head.

[decl] :: t{+[FIN|INF].-IMP.-INV.[+ELLIP|+EMPH|+NOT|-DO].x}= +negs{+NONE}?  
+wh{+OVERT}? c{DECL.NULL.SUBORD.x.y} % Subordinate clause null C head.

that :: t{+FIN.-IMP.-INV.-RREL.x}= +wh{+[REL|-NOM]}?  
c{DECL.OVERT.SUBORD.THAT.x} % Overt declarative subordinate clause C head.

### A.7.2 Imperatives

The analysis for imperatives needs revising as it currently uses both a [imp] mod head  
and an [imp] C head, but should only use the C head given that imperative is usually

regarded as a force rather than a mood.

[mod] :: t{+[EMPH|IMP].-INV.-MD.[+EMPH|+NOT|-DO].[+EMPH|+NOT|-ELLIP].x}= +TOP{-LINV}! +WH{+PURP}! +neg{+NONE}? +foc{+NONE}? c{MAIN.NULL.x}

[mod] :: lv{+BARE.-ELLIP.x}= +pol! mod{+[2PL|2SG].IMP.x}

[pro-d] :: D{2PL.2SG.DEF.IMP.NOMOD.NULL.PRO} -case{2PL.2SG.IMP.NULL.PRO}  
%cased null pronominal used in imperatives without overt subject.

[pres] :: >mod{+[AUX|-ELLIP].x}= +CASE{-EXTRAP.x}  
+pol! t{FIN.IND.NULL.PRES.x}

you :: D{2PL.2SG.DEF.OVERT.PRD.PRO} -case{2PL.2SG.ACC.NOM.OVERT}

### A.7.3 Exclamatives

[excl] :: t{+FIN.-IMP.-INV.x}= +WH{+OVERT.[+HOW|WHAT].-ECHO.-PRO.-REL}  
+negs{+NONE}? +foc{+NONE}? +top{+NONE}? c{EXCL.MAIN.NULL.x}

### A.7.4 Conditionals

if :: t{+FIN.-IMP.-INV.-RREL.x}= +pol! ≈lv{+OVERT}

if :: t{+FIN.-IMP.-INV.-RREL.x}= +pol! ≈lv{+OVERT} -top{IF} % Used when *if* clause undergoes topicalization and appears in the left periphery (cannot be base generated there because it can still modify an embedded clause: *if you go there, they say that you will never come back.*).

### A.7.5 Interrogatives

[int] :: >t{+AUX.+FIN.-IMP.-INV.x}= +wh{+NONE}? +negs{+NONE}?  
+foc{+NONE}? +TOP{+IF}! +pol! c{INT.MAIN.NULL.x} % Main clause yes-no  
interrogative C head.

whether :: t{-IMP.-INV.-RREL.[+EMPH|-DO].[+FIN|INF].x}= +wh{+ECHO}? +foc{+NONE}?  
+top{+NONE}? +pol! c{INT.OVERT.SUBORD.x} % Subordinate clause yes-no in-  
terrogative C head.

[int] :: >t{+FIN.-IMP.-INV.[+EMPH|-DO].x}= +WH{+NOM.+OVERT.-ECHO.-REL.y}  
 +negs{+NONE}? +foc{+NONE}? +TOP{+IF}! +pol! c{INT.MAIN.NULL.x.y} %  
 Main clause wh-subject question C head.

[int] :: >t{+AUX.+FIN.-IMP.-INV.x}= +WH{+OVERT.-ECHO.-NOM.-REL}  
 +negs{+NONE}? +foc{+NONE}? +TOP{+IF}! +pol! c{INT.MAIN.NULL.x} %  
 Main clause wh-object question C head.

[int] :: t{-IMP.-INV.-RREL.[+EMPH|-DO].x}= +WH{+OVERT.-ECHO.-REL}  
 +negs{+NONE}? +foc{+NONE}? +top{+NONE}? +pol! c{INT.NULL.SUBORD.x}  
 % Embedded wh-question C head.

what/which :: n{+OVERT.x}= D{INDEF.NEUT.NOM.OVERT.WH.x}  
 -case{NEUT.NOM.OVERT.WH.x} -wh{NOM.OVERT.x} % Nominative version of  
 wh determiner.

what/which :: n{+OVERT.x}= D{ACC.GEN.INDEF.NEUT.OVERT.WH.x}  
 -case{ACC.GEN.NEUT.OVERT.WH.x} -wh{ACC.OVERT.x} % Accusative version  
 of wh determiner.

who :: D{3SG.INDEF.MASC.NOM.OVERT.PRO.WH}  
 -case{3SG.MASC.NOM.OVERT.PRO.WH} -wh{+3SG.+MASC.NOM.OVERT.PRO}  
 % 3rd singular, masculine, nominative pronoun *who*

who :: D{3SG.ACC.FEM.INDEF.OVERT.PRO.WH}  
 -case{3SG.ACC.FEM.OVERT.PRO.WH} -wh{+3SG.+FEM.ACC.OVERT.PRO} % 3rd  
 singular, feminine, accusative version of *who*

why :: ≈lv -wh{OVERT.PURP.VMOD} % Purposive wh adverbial

where :: ≈lv -wh{LOC.OVERT.VMOD} % Locative wh adverbial

### A.7.6 Echo questions

[int] :: t{-IMP.-INV.-RREL.[+EMPH|-DO].x}= +wh{+ECHO.+OVERT.-REL}  
 +negs{+NONE}? +foc{+NONE}? +TOP{+IF}! +pol! c{INT.MAIN.NULL.x} %  
 Main clause [int] head which triggers covert wh movement for echo questions (You  
 wondered whether *who* was coming to the party?!)

who :: D{3SG.ACC.INDEF.MASC.OVERT.PRO.WH}

-case{3SG.ECHO.MASC.NOM.OVERT.PRO.WH}

-wh{+3SG.+MASC.ECHO.NOM.OVERT.PRO} % special echo version of *who* (the ECHO property allows it to escape *if/whether* wh-islands created by +wh{+ECHO}? on *if/whether*).

See also section A.2.10 for ‘perception gerunds’ such as *he was seen running away*.

## A.8 Adverbs

now :: adv{OVERT.TMP}

slowly :: adv{MNR.OVERT}

slowest/most :: adv{ADMOD.OVERT.RBS}

slower :: adv{MNR.OVERT.RBR}

just/only/even/also :: adv{ADMOD.FOC.OVERT}

however/nevertheless :: adv{DIS.OVERT}

[adjunctizer] :: adv{+[MNR|RBR|RBS].-NOMOD}=  $\approx_v$ {-NOMOD}

[adjunctizer] :: adv{+TMP.-NOMOD}=  $\approx_t$ {-NOMOD}

## A.9 More A'-movement

### A.9.1 Relative clauses

who :: n{+3PL.+OVERT.x}= D{3PL.INDEF.NOM.OVERT.WH.x}

-case{3PL.NOM.OVERT.WH} -wh{+3PL.-FEM.-MASC.NOM.OVERT.x} % Special wh determiner version of *who* to implement promotion analysis (Bhatt, 2002) of restrictive relatives. Appositive relatives adopt a more traditional adjunction approach and just use the same wh pronouns as wh-interrogatives.

who :: n{+FEM.+OVERT.x}= D{3SG.FEM.INDEF.NOM.OVERT.WH}

-case{3SG.FEM.NOM.OVERT.WH} -wh{+3SG.+FEM.NOM.OVERT.x}

who :: n{+MASC.+OVERT.x}= D{3SG.ACC.INDEF.MASC.OVERT.WH}

-case{3SG.ACC.MASC.OVERT.WH} -wh{+3SG.+MASC.ACC.OVERT.x}

when :: n{+OVERT.+REL.+TMP.x}= ≈t{+OVERT} -wh{OVERT.x} % *There are times when she must show a little emotion.*

[rel] :: c{+THAT.-EXTRAP.-FOC}= +foc{+NONE}! +top{+NONE}!  
+WH{+OP.+OVERT.x} c{NOMOD.NULL.RELAT.SUBORD.x} % Relative clause C head for *that*-restrictive relatives.

[rel] :: c{+DECL.+SUBORD.-EXTRAP.-FOC.-INF.-RREL.-THAT}= +foc{+NONE}!  
+top{+NONE}! +WH{+OVERT.-ECHO.-[NOM|OP].x}  
c{NOMOD.NULL.RELAT.SUBORD.x} % Relative clause head for appositive and non-*that* restrictive relatives (different C heads for *that* and non-*that* relatives are needed to enforce anti-*that* trace effects).

[rel] :: c{+FOR.-EXTRAP.-FOC}= +foc{+NONE}! +top{+NONE}!  
+WH{+OP.+OVERT.-NOM.x} c{NOMOD.NULL.RELAT.SUBORD.x} % Relative clause head for for-to INF relatives (*a pen for me to write with, there's no way for them to lose*). +OP restriction ensures the wh item in spec-CP is null: *\*a pen which for to write with. \*a pen with which for to write.*

[rel] :: c{+INF.-EXTRAP.-FOC.-FOR}= +foc{+NONE}! +top{+NONE}!  
+case{+ACC.+OVERT.+REL}! +WH{+OVERT.+[OP|P|PRO|VMOD].x}  
c{NOMOD.NULL.RELAT.SUBORD.x} % Relative clause head for non-for-to INF relatives, including in tough movement constructions (*a pen to write with, that got hard to take*). +[OP|P|PRO|VMOD] restriction allows overt wh item only if governed by preposition: *a pen \*(with) which to write.*

[rel] :: c{+DECL.+OVERT.+RREL.+SUBORD.-EXTRAP.-FOC}= +foc{+NONE}!  
+top{+NONE}! +WH{+[NOM|VMOD].+[OP|VMOD].+[REL|VMOD].-ECHO.x}  
c{NOMOD.NULL.OVERT.RELAT.RREL.SUBORD.x} % Reduced relative C head (*The exact refund will be based on actual collections made*).

[adjunctizer] :: c{+RELAT.-OP.-RREL.[+LOC|-VMOD].x}= +N{+NONE}!  
≈d{+OVERT.-NOMOD.-OP.-REL.[+S|-PRO].x} % Used to adjoin [rel] CP to DP for appositive relative clauses.

[tense] :: >lv{+PASS.-ELLIP.x}= t{FIN.IND.NULL.PAST.PRES.RREL.x} % T head of passive reduced relative with controlled null subject (*When referred to the questions that matched, he said it was coincidental*).

[tense] :: >lv{+PASS.[+AUX]-ELLIP}.x}= +CASE{+NOM.+OVERT.-EXTRAP.x}  
 t{FIN.IND.NULL.PAST.PRES.RREL.x} % T head of reduced passive relative with  
 overt subject (*Dodge reported an increase in construction contracts awarded in Septem-  
 ber*).

[tense] :: >lv{+PROG.[+AUX]-ELLIP}.x}= +CASE{+NOM.+OVERT.-EXTRAP.x}  
 t{FIN.IND.NULL.PAST.PRES.RREL.x} % T head for reduced progressive relative  
 clause (*You've got two champions sitting right before you*).

[relativizer] :: n{+OVERT.-JJR.-NAME.-NUM.-RELAT.-SH.x}=  
 n{NOMOD.NULL.REL.x} -n{EDGE.RELAT.x} % Adds -n licensee to NP, moving  
 it to left periphery of restrictive relative clause.

[wh] :: n{+OVERT.+REL.x. EDGE}= +N{y} D{INDEF.NOMOD.NULL.OP.WH.x}  
 -case{ACC.GEN.OP.WH.x} -wh{OP.x} -n{y} % Null nominative wh determiner used  
 in restrictive relative clauses without overt wh-item.

[wh] :: n{+OVERT.+REL.x. EDGE}= +N{y} D{INDEF.NOMOD.NULL.OP.WH.x}  
 -case{NOM.OP.WH.x} -wh{NOM.OP.x} -n{y} % Accusative version of the above.

[wh] :: n{+OVERT.+[LOC|MNR|PURP|TMP].x. EDGE}= +N{y} ≈lv{-NOMOD}  
 -wh{OP.x} -n{y} % Used when the relativized NP is an adverbial (*The place where he  
 sleeps at night*).

[det] :: c{+FIN.+INT.+OVERT.+SUBORD.-EXPL}= +self{+NONE}!  
 =d{+FIN.+INT.+WH.+WHAT} lv{+3SG.AUX.BE.INTRANS.LV.OVERT.PAST} %  
 Maps CP relative clause into DP free relative clause (*I like what you say*).

c{+OVERT.+VMOD.-GER.-INF}= +log! +N{+NONE}? ≈lv{-NOMOD.-PASS} %  
 Used to adjoin adverbial clause, including adverbial free relative, to vP (*I finished how  
 you began*).

## A.9.2 Clefts

[relativizer] :: d{+3SG.+OVERT.+PRO.+[DEM|NEUT].+[DEM|PERS].-POSS.x}= +case  
 n{3SG.FEM.IT.LOC.MASC.NEUT.NOMOD.NULL.OVERT.REL.TMP} -n{EDGE.IT.x}  
 % Special relativizer of *it* in clefts, so *It is the total relationship that is important* is  
 treated as being derived from something like *It that is important is the total relation-  
 ship*, with the *that* clause being extraposed.

*it* :: D{3SG.ACC.DEF.NEUT.NOM.OVERT.PERS.PRD.PRO}  
 -case{3SG.ACC.NEUT.NOM.OVERT.PRO} % Just standard personal pronoun *it* used  
 in clefts.

[det] :: c{+EXTRAP.+IT.+RELAT}= +tough{+NONE}! +N{x}  
 D{CLEFT.EXPL.NOMOD.RELAT.x} -case{3SG.ACC.NOM.OVERT} % [det] head  
 selecting the rightward moving relative clause as its complement and transforming it  
 to a DP so it can move to spec-TP.

[extraposer] :: c{-EXTRAP.-FOC.-SHELL.-TOPIC.x}= c{EXTRAP.NOMOD.NULL.x}  
 t{EDGE} % Extraposes the relative clause to the end of the sentence.

### A.9.3 Pseudo-clefts

*was* :: c{+DECL.+FIN.+OVERT.+THAT.-EXPL}= +self{+NONE}?  
 =d{+DECL.+FIN.+WH} lv{+3SG.AUX.BE.INTRANS.LV.OVERT.PAST} % *What Chom-*  
*sky said was that syntax involves transformations.*

*was* :: c{+FIN.+INT.+OVERT.+SUBORD.-EXPL}= +self{+NONE}!  
 =d{+FIN.+INT.+WH.+WHAT} lv{+3SG.AUX.BE.INTRANS.LV.OVERT.PAST} *What*  
*Chomsky wondered was whether syntax involves transformations.*

### A.9.4 Topicalization

[topicalizer] :: d{+OVERT.-DECL.-EXPL.-INF.-OP.[+FREL|-REL].[+FREL|-WH].x}=  
 +case{y} D{NOMOD.TOP.x} -case{TOP.y} -top{OVERT.y} % *That kind of behaviour,*  
*he would not tolerate.*

[topicalizer] :: p{+VMOD.-DAT}= ≈lv{-NOMOD} -top % *With that one left move,*  
*the game was over.*

[decl] :: t{+FIN.-IMP.-INV.-RREL.-TOPIC.[+ELLIP|+EMPH|+NOT|-DO].x}=  
 +foc{+NONE}? +wh{+NONE}? +negs{+NONE}? +TOP{-LINV}  
 c{DECL.MAIN.NULL.x} % Declarative C head attracting topic.

### A.9.5 Quotative inversion and discontinuous quotatives

[past] :: >lv{+PAST.+REPORT.-BE.[+AUX|-ELLIP].x}= +CASE{+NOM.+OVERT.-  
 EXTRAP.x} t{FIN.IND.INV.NULL.PAST.x} % past tense T head triggering V-to-T



movement for quotative inversion (treated as remnant V2).

[focalizer] :: c{-REL.-RELAT.[+DECL|+INT|-TOPIC].[+TEXTRAP|-EXTRAP].x}=  
c{FOC.NOM.NOMOD.x} -foc

[decl] :: >t{+FIN.+INV.+REPORT.-IMP.[+ELLIP|+EMPH|+NOT|-DO]}=  
+FOC{-COORD.-SO} +wh{+NONE}? +negs{+NONE}? c{DECL.MAIN.x} % Declar-  
ative C head triggering T-to-C movement.

[extraposer] :: t{-ELLIP.-EXTRAP.x}= t{EXTRAP.TEXTRAP.x}  
v{+[IREPORT|REPORT]} % *Funny thing* is adjoined to the subordinate CP, which  
undergoes quotative inversion to spec-CP of main clause, then the subordinate TP un-  
dergoes rightward movement resulting in a discontinuous quotation (“*Funny thing,*”  
*says the kicker, “both these candidates are named Rudolph Giuliani.”*)

### A.9.6 Negative Inversion

[decl] :: >neg{+FIN.+INV.+REPORT.+UPPER.-IMP.-P}= +FOC{-COORD.-SO}  
+wh{+NONE}? +negs{+NONE}? c{DECL.MAIN.x} % [decl] C head attracting  
focused element and triggering T-to-C head movement (*Never again would he tolerate  
that kind of behaviour*).

never :: adv{+TMP.-NEG}= negs{OVERT.TMP} -foc{NEG}

again :: adv{OVERT.TMP}

### A.9.7 Heavy DP shift

[extraposer] :: d{+OVERT.-DECL.-EXPL.-EXTRAP.-FL.-FOC.-INF.-OP.-PRO.-REL.-  
SELF.-TOP.x}= +case{y} d{EXTRAP.NOMOD.x} -case{EXTRAP.y} t % *We are  
prepared to pursue aggressively completion of this transaction.*

### A.9.8 Tough movement

tough/hard/difficult/easy :: c{+RELAT.+TOUGH}= +TOUGH adj{OVERT.PRD.TOUGH}

[op] :: d{+OVERT.-EXPL.-NOMOD.-OP.-WH.x}= +case{+OVERT.y}  
D{NOMOD.NULL.OP.x} -case{ACC.OVERT} -wh{OP.OVERT.TOUGH} -tough D{x}  
-case{y}

[op] :: d{+OVERT.+WH.-EXPL.x}= +case{+OVERT.y} +wh{+OVERT.z}  
 D{NOMOD.NULL.OP.x} -case{ACC.OVERT} -wh{OP.OVERT.TOUGH} -tough D{x}  
 -case{y} -wh{z} % Used when tough movement feeds further wh-movement (*Networks develop images in peoples' minds that aren't easy to change, which sonata is easy to play on this violin*).

### A.9.9 The violin-sonata paradox

Note that the grammar correctly generates 387-392 and correctly blocks 393. The reader can inspect the derivations of 387-392 using the Autobank system (just do a search for the word 'violin' in the seed set; the examples with *it* involve *it*-extraposition); here we will just focus on the contrast between 392 and 393. Note that the analysis of tough movement was presented in full in section 6.4.2.1.

- (387) It is easy to play sonatas on the violin.  
 (388) The violin is easy to play sonatas on.  
 (389) The sonata is easy to play on this violin.  
 (390) What violin is it easy to play sonatas on?  
 (391) What sonatas is it easy to play on the violin?  
 (392) What violin is this sonata easy to play on?  
 (393) \*What sonatas is this violin easy to play on?

Chomsky (1977) argues that 393 (which involves crossing dependencies) is ill-formed because it violates a wh-island owing to the presence of a null wh-operator (which has moved from the object position of *on*) in spec-CP of the infinitival clause. Chomsky assumes the following structure for 388 to which wh-movement must apply to generate 393 (Chomsky represents the hypothesised null wh-operator as *which* and the null infinitival complementizer as *for*); this shows that *sonatas* is indeed inside a wh-island under Chomsky's assumptions.

- (394) [The violin]<sub>i</sub> is easy [<sub>S'</sub> [(which) for] [<sub>S</sub> PRO to play sonatas on t<sub>i</sub>]].

In MGBank, the null wh-operator is treated as a trace of A-movement that follows the A'/wh-movement step to spec-CP, i.e. tough movement in MGBank involves so-called *improper movement* (see section 6.4.2.1). 393 is therefore blocked because it

would involve an SMC violation: there will be two -wh licensees simultaneously active at a certain point in the derivation, one on the tough mover *this violin* as it moves to spec-CP of the infinitival clause, and one on the wh phrase *what sonatas* which is trying to escape that CP. Assuming *on this violin* to be generated as a complement of *play*, 392 would involve nested, rather than crossing dependencies, but this does not help as it is still the case that there would have to be two -wh licensees simultaneously active in the derivation. This example is also problematic under Chomsky's wh-island approach. To get around this, Chomsky (1977) argues that 389, from which 392 is derived, has two possible structures, which are shown below.

- (395) a. [The sonata]<sub>i</sub> is easy [<sub>S'</sub> [(which) for] [<sub>S</sub> PRO to play t<sub>i</sub> on this violin]].  
 b. [The sonata]<sub>i</sub> is [<sub>AP</sub> easy [<sub>S'</sub> [(which) for] [<sub>S</sub> to play t<sub>i</sub>]]] on this violin.

In 395b, the PP is adjoined to the VP headed by *is* and is therefore outside of the infinitival clause meaning that *which violin* would not have to cross the wh-island at all on its way to the matrix spec-CP in 392. This would also avoid the SMC violation in the MG context. However, Takami (1992, p.173) argues that this cannot be quite correct because applying topicalisation to the adjP in 389 results in the PP being carried along, showing that *easy to play on her violin* forms a constituent.

- (396) My teacher told me that this sonata might be easier to play on her violin than on mine and easier to play on her violin it was.

One could try to argue that the PP is carried along here because 396 is derived from the structure in 395a rather than that in 395b. However, leaving the PP in place appears to my ears at least to result in a degraded sentence.

- (397) ??My teacher told me that this sonata might be easier to play on her violin than on mine and easier to play it was on her violin.

For this reason, MGbank treats the PP in 393 as being adjoined to the adjP headed by *easy*, rather than to the VP headed by *is*. This still avoids the SMC violation, maintains the intended semantics, and better accounts for the the topicalization data above because now the PP cannot be stranded when the *easy* adjP is topicalized.

### A.9.10 Pied-piping heads

These heads are used when the wh-item is embedded inside a larger XP that moves with it.

[pied-pipe] :: d{-EXPL.-OP.-WH.x}= +case{y} +wh{+PRO.-OP.z} d{WH.x} -case{y} -wh{PIED.z} % *The swap, details of which were disclosed last Thursday, shows the dynamism of France's state.*

[pied-pipe] :: p{-PIED.x}= +wh{-OP.y} p{PIED.x} -wh{y} % *To which country did he sail.*

[pro-v] :: =d{+[IMP|OVERT].-EXCL.-EXPL.-EXTRAP.[+OVERT|-OP]}  
lv{BARE.ELLIP.INTRANS.NULL.PERF.PRO.PROG.TRANS.x} % Ellipsis not currently handled properly; VP ellipsis just uses null verbal heads (*You either believe Seymour can do it, or you don't*).

[pro-v] :: d{+OVERT.[+TOP|WH].-EXCL.-EXPL.-EXTRAP}=  
+CASE{+ACC.+OVERT} +WH? +TOP?  
=d{+[IMP|OVERT].-EXCL.-EXPL.-EXTRAP.[+OVERT|-OP]}  
lv{BARE.ELLIP.NULL.PERF.PRO.PROG.TRANS.x} % Used when the object escapes the ellipsed VP (*The people who Jack likes and Mary does not have arrived*).

## A.10 The nominal domain

### A.10.1 Nouns, names and honorifics

hat :: n{3SG.OVERT}

picture :: p{+NCOMP}= n{3SG.OVERT} % (*picture of Mary*)

claim/sign :: c{+DECL}= +wh{+ECHO}? +foc{+NONE}! +top{+NONE}!  
n{3SG.HEAVY.OVERT} (*Economists consider it a sign that inflationary pressures are abating.*)

man :: n{3SG.MASC.OVERT} % Agreement features percolate up to DP level and allow agreement with same gender/person/number reflexive.

woman :: n{3SG.FEM.OVERT}

jack :: n{3SG.MASC.NAME.OVERT}

mary :: n{3SG.FEM.NAME.OVERT}

microsoft :: n{3SG.NEUT.NAME.OVERT}

mr :: n{3SG.MASC.NAME.OVERT}

mrs :: n{3SG.FEM.NAME.OVERT}

dr :: n{3SG.NEUT.NAME.OVERT}

Note that in MGbank, when two proper noun NPs are compounded via adjunction, the leftmost proper noun is always treated as the head. This is precisely the opposite to the situation for regular NP compounds, such as *table sugar*, where the leftmost noun is the modifier. The reason MGbank treats proper nouns differently is that the leftmost proper noun is invariably the one with the gender that defines the entire NP, because first names and honorifics appear leftmost in the NPs in which they appear. Making these items the heads ensures that the gender feature will percolate up to the DP level so that gender agreement can apply between this DP and a reflexive anaphor.

### A.10.2 Determiners

the :: n{+OVERT.-REL.x}= D{ACC.DEF.GEN.NOM.OVERT.PR.D.x}  
-case{ACC.DEF.GEN.NOM.OVERT.x}

the :: q{x}= D{ACC.DEF.NOM.OVERT.PR.D.x} -case{ACC.DEF.NOM.x} % *The two sides hadn't met since October, The many differences of opinion created an insurmountable rift.*

every :: n{+OVERT.+[3SG|COORD].-REL.-UNI.[+TMP|-NAME].x}=  
D{ACC.DEF.GEN.NOM.OVERT.PR.D.UNI.x} -case{ACC.DEF.GEN.NOM.OVERT.x}  
% UNI property included to differentiate this determiner from other definites for possible implementation of Quantifier Raising in the future.

this/that :: n{+3SG.+OVERT.-REL.x}= D{ACC.DEF.DEM.GEN.NOM.OVERT.PR.D.x}  
-case{ACC.DEF.GEN.NOM.OVERT.x}

these/those :: n{+3PL.+OVERT.-NAME.-REL.x}= D{3PL.DEF.DEM.OVERT.PR.D.x}  
-case{ACC.DEF.GEN.NOM.OVERT.x}

these/those :: q{+3PL.+OVERT.-NAME.x}= D{3PL.DEF.DEM.OVERT.PR.D.x}  
-case{ACC.DEF.GEN.NOM.OVERT.x} % *Those four candidates were exemplary.*

a/some :: n{+3SG.+OVERT.-NAME.-REL.x}=  
D{ACC.GEN.INDEF.NOM.OVERT.PR.D.x}

-case{ACC.GEN.NOM.OVERT.x}

several/some :: n{+3PL.+OVERT.-NAME.-REL.x}=

D{ACC.GEN.INDEF.NOM.OVERT.PRD.x} -case{ACC.GEN.NOM.OVERT.x}

two/ten/many :: n{+OVERT.-REL.x}= q{OVERT.x}

such :: d{+3SG.+INDEF.+OVERT.-EXPL.x}= +case{y} D{3SG.DEF.OVERT} -case{y}

all/half :: d{+3SG.+OVERT.+[DEF|NULL].-EXPL.-UNI.[+DEM|-PRO].x}= +case{y}

D{Q.x} -case{y} % All the candidates were exemplary.

any/much :: n{+OVERT.-NAME.-REL.x}= D{ACC.GEN.INDEF.NOM.OVERT.PRD.x}

-case{ACC.GEN.NOM.OVERT.x} -pol{OVERT}

no :: n{+OVERT.-NAME.-REL.x}= D{ACC.GEN.INDEF.NOM.OVERT.PRD.x}

-case{ACC.GEN.NOM.OVERT.x} -negs{OVERT}

[det] :: n{+OVERT.-NAME.-REL.x}= D{ACC.GEN.INDEF.NOM.NULL.PRD.x}

-case{ACC.GEN.INDEF.NOM.x} % For bare NPs with no overt determiner (*Sugar tastes good*)

[det] :: n{+NAME.+OVERT.-REL.x}= D{ACC.DEF.GEN.NOM.NULL.PRD.x}

-case{3SG.ACC.DEF.GEN.NOM.x}

[det] :: q{+OVERT.x}= D{ACC.DEF.INDEF.NOM.NULL.PRD.x}

-case{ACC.DEF.NOM.x}

### A.10.3 Pronouns

i :: D{1SG.DEF.NOM.OVERT.PERS.PRD.PRO} -case{1SG.NOM.OVERT.PRD}

me :: D{1SG.ACC.DEF.OVERT.PERS.PRD.PRO} -case{1SG.ACC.OVERT.PRD}

we :: D{1PL.DEF.NOM.OVERT.PERS.PRD.PRO} -case{1PL.NOM.OVERT.PRD.PRO}

D{1PL.ACC.OVERT.PERS.PRD.PRO} -case{1PL.ACC.OVERT.PRD}

he :: D{3SG.DEF.MASC.NOM.OVERT.PERS.PRD.PRO}

-case{3SG.MASC.NOM.OVERT.PRD.PRO}

they :: D{3PL.DEF.NOM.OVERT.PERS.PRD.PRO} -case{3PL.NOM.OVERT.PRD.PRO}

us :: D{3PL.ACC.DEF.OVERT.PERS.PRD.PRO} -case{3PL.ACC.OVERT.PRD.PRO}

this/that :: D{3SG.ACC.DEF.DEM.NOM.OVERT.PRD.PRO}

-case{3SG.ACC.NOM.OVERT.PRO}

these/those :: D{3PL.ACC.DEF.DEM.NEUT.NOM.OVERT.PRD.PRO}

-case{3PL.ACC.NEUT.NOM.OVERT.PRO}

some :: D{ACC.GEN.INDEF.NOM.OVERT.PRD.x} -case{3PL.ACC.GEN.NOM.OVERT}

something :: D{3SG.ACC.GEN.INDEF.NEUT.NOM.OVERT.PRD.PRO}

-case{3SG.ACC.GEN.NEUT.NOM.OVERT.PRO}

anything :: D{3SG.ACC.GEN.INDEF.NEUT.NOM.OVERT.PRD.PRO}

-case{3SG.ACC.GEN.NEUT.NOM.OVERT.x} -pol{OVERT}

one :: n{3SG.OVERT.PRO}

#### A.10.4 Possessives

The analysis of possessives is shortly to be revised so that 's is generated directly in D rather than moving there from little n; N will undergo N-to-n head movement (Radford, 2004). Below are the categories under the current old analysis. ln = light noun, or little n.

's :: n{-REL.x}= +self! =d{+GEN.-ANA.-EXPL.-PERS.[-DEM|-PRO].[-POSS|-WH]}

ln{ACC.OVERT.POSS.S.x}

[ 's ] :: >n{+OVERT.+PRO.x}= =d{+GEN.+PRO.+[POSS|SPOSS].-EXPL.[+OVERT]-OP} ln{ACC.NULL.S.x} % Used with strong genitive pronouns (*mine, yours, etc*).

[ 's ] :: n{+OVERT.-PASS.-PRO.-REL.x}= +self! =d{+POSS.+[PRO|WH].-EXPL.-S.[+OVERT]-OP} ln{ACC.NULL.S.x} % Used with weak genitive pronouns (*my, your, etc*).

my :: D{+1SG.1SG.GEN.OVERT.PERS.POSS.PRO}

-case{1SG.GEN.OVERT.PERS.POSS.PRO}

mine :: n{3SG.GEN.OVERT.PERS.PRO.SPOSS} D{1SG.GEN.OVERT.PERS.PRO.SPOSS}

-case{3SG.ACC.GEN.NOM.OVERT.PERS.SPOSS}

[gen] :: >ln{+S.x}= +CASE{+GEN.+OVERT} +wh{+NONE}!

$D\{\text{ACC.DEF.GEN.INDEF.NOM.NULL.PR.D.x}\} -\text{case}\{\text{ACC.DEF.GEN.INDEF.NOM.x}\}$

### A.10.5 Prepositions

to ::  $d\{+\text{OVERT}.\text{-EXPL}.\text{-PHI}\} = +\text{case}\{+\text{ACC}.\text{+OVERT}\} p\{\text{OVERT.PR.D.VCOMP}\}$   
 % Prepositional complement of verb.

of ::  $d\{+\text{OVERT}.\text{-EXPL}.\text{-PHI.x}\} = +\text{case}\{+\text{ACC}.\text{+OVERT}\} p\{\text{NCOMP.OVERT.PR.D.x}\}$   
 % Prepositional complement of noun.

in ::  $d\{+\text{OVERT}.\text{-EXPL}.\text{-PHI}\} = +\text{case}\{+\text{ACC}.\text{+OVERT}\}$   
 $p\{\text{LOC.NCOMP.NMOD.OVERT.PR.D.VCOMP.VMOD}\}$  % Locative adverbial preposition.

by ::  $d\{+\text{OVERT}.\text{-EXPL}\} = +\text{case}\{+\text{ACC}.\text{+OVERT}\}$   
 $p\{\text{NMOD.OVERT.PR.D.TMP.VMOD}\}$  % Temporal adverbial preposition.

### A.10.6 Gerunds

[gerund] ::  $lv\{+\text{OVERT}.\text{+PROG}.\text{-ELLIP.x}\} = +\text{CASE}\{+\text{ACC}.\text{-EXTRAP}\} t\{\text{ACC.GER.x}\}$   
 % Gerund T head checking accusative case for clausal gerunds (*Jack likes Mary helping*).

[gerund] ::  $lv\{+\text{OVERT}.\text{+PROG}.\text{-ELLIP.x}\} = t\{\text{CONT.GER.x}\}$  % Gerund T head not checking case for control into gerunds (*Jack likes helping*).

[det] ::  $t\{+\text{OVERT}.\text{+PROG}.\text{-ELLIP}.\text{-RREL}.\text{-SO}.\text{-TOP.x}\} =$   
 $d\{\text{3SG.ACC.DEF.GEN.INDEF.NOM.NOMOD.NULL.PR.D.x}\}$   
 $-\text{case}\{\text{3SG.ACC.DEF.GEN.INDEF.NOM.x}\}$  % [det] head converting gerund TP (clausal or control) into DP with -case.

after/while/though ::  $t\{+\text{CONT}.\text{+GER}.\text{-ACC}.\text{-NOMOD}.\text{-RREL.x}\} = +\text{wh}\{-\text{VMOD}\}?$   
 $c\{\text{OVERT.VMOD.x}\}$  % *I'm really wiped out after walking five hours.*

[adjunctizer] ::  $c\{+\text{GER}.\text{+OVERT}.\text{-ELLIP}.\text{-FOR}.\text{-NOMOD}\} = +\text{log!} \approx lv\{-\text{NOMOD}\}$   
 % Rights adjoins gerund CP to vP.

without ::  $t\{+\text{GER}.\text{-NOMOD}.\text{-RREL.x}\} = +\text{pol!} +\text{wh}\{-\text{VMOD}\}?$   $c\{\text{OVERT.VMOD.x}\}$   
 % Gerund clause complementizer that licenses polarity item (*Investors can essentially buy the funds without paying any premium*).



[topicalizer] ::  $c\{+OVERT.+[GER|INF|VMOD].-ELLIP.-EXTRAP.-FOR\}= +log!$   
 $\approx lv\{-NOMOD.-PASS\}$  -top % Right adjoins gerund adverbial clause with overt complementizer to vP, the subject ATB moves from gerund clause to main clause, then the gerund clause undergoes remnant topicalization movement (*Without having looked at them, I can't possibly make a recommendation*).

[adjunctizer] ::  $t\{+GER.+PROG.-ELLIP.-EXTRAP.-NOMOD.-RREL\}= +top\{+NONE\}?$   
 $\approx lv\{-NOMOD.-PASS\}$  % Right adjoins bare progressive TP to vP with ATB movement of the subject for adjunct control (*He joins November 5th, dissolving his consulting firm.*)

[topicalizer] ::  $t\{+OVERT.+[GER|PASS].+[PASS|PROG].-ELLIP.-EXTRAP.[+PASS|-RREL].x\}= \approx lv\{-NOMOD\}$  -top % Right adjoins bare progressive TP to vP then causes the TP to be topicalized (another case of remnant movement as the subject first ATB moves out of the adjunct: *Hoping to stay ahead of the pack, the company is emphasizing innovation*).

## A.11 Adjectives

adj{OVERT.PRD}

adj{JJS}

adj{CMP.JJR.OVERT.PRD}

[adjunctizer] ::  $adj\{+JJR\}= \approx n\{+OVERT.-NOMOD\}$  % Right adjoins comparative adjP to NP (*Some funds are posting yields far higher than the average*.)

[adjunctizer] ::  $adj\{+OVERT.-NOMOD\}= n\{+OVERT.-NOMOD.-REL\} \approx$

## A.12 Coordination

### A.12.1 Basic coordinators

and ::  $p\{+DAT.+OVERT.-PASS\}= =p\{+DAT.+OVERT.-PASS.x\} D\{3PL.COORD.PRD.x\}$   
 $-case\{3PL.ACC.GEN.NOM.OVERT\}$  % DP coordinator, but actually coordinates PPs headed by null [dat] head to eliminate the case of the coordinated DPs.

[dat] :: d{+OVERT.-DECL.-EXPL.-INF.-PHI.-PROG}= +case{x}  
 p{DAT.NOMOD.NULL.x}

and :: c{+DECL.+OVERT.+SUBORD}= =c{+DECL.+OVERT.+SUBORD.x}  
 c{COORD.x}

and :: adj= =adj{x} adj{COORD.OVERT.x}

or :: adj= =adj{x} adj{COORD.DISJ.OVERT.x}

nor :: adj= =adj{x} adj{COORD.DISJ.NEG.OVERT.x}

and :: q= =q n{+OVERT.-REL.x}= q{COORD.LH.OVERT.x} % Lexical head coordinator of QPs (*I've seen one or two men die, bless them*)

### A.12.2 ATB head movement and argument cluster coordinators

and :: t{+AUX.+FIN}=^ =t{+AUX.+FIN.-ELLIP.x}^ t{COORD.EXCORP.x} % Coordinator of TPs with excorporation head movement, for across-the-board head and phrasal movement (*Who does Jack love and Mary hate?*)

and :: lv{+OVERT}=^ =lv{+OVERT.-ELLIP.x}^ lv{COORD.EXCORP.OVERT.x} % vP coordinator with excorporation head movement, for argument cluster coordination (*This has both made investors uneasy and the corporations more vulnerable*).

### A.12.3 Correlative focus coordinator particles

either :: part{3SG.COORD.DISJ.FOC} -foc{COORD.EDGE.OVERT}

neither :: part{3PL.COORD.DISJ.FOC.NEG} -foc{COORD.EDGE.OVERT}

both :: part{3PL.COORD.FOC} -foc{COORD.EDGE.OVERT}

[correlativizer] :: >lv{+COORD.+EXCORP.-CORREL.-DISJ.x}=  
 =part{+COORD.+FOC.-DISJ} lv{CORREL.x} % Creates doubled constituent consisting of a coordinate vP and a correlative focus particle (*both, either, neither, etc*).

[correlativizer] :: d{+COORD.-CORREL.-DISJ.x}= +case{y} =part{+COORD.+FOC.-DISJ} D{CORREL.x} -case{y} % Creates doubled constituent consisting of a coordinate DP and a correlative focus particle (*both, either, neither, etc*).

[foc] :: d{+OVERT.x}= +case{+OVERT.y} +FOC{+COORD.+OVERT} D{x} -case{y}  
 % Focus head at edge of DP attracts correlative focus particle (*Pete said that Jack likes either apples or bananas (but I don't remember which)*).

[foc] :: lv{+OVERT.x}= +FOC{+COORD.+OVERT} lv{x} % *This has both made investors uneasy and the corporations more vulnerable, Pete said that Jack either likes apples or bananas (but I don't remember which), Pete either said that Jack likes apples or bananas (but I don't remember which)*.

[foc] :: t{+COORD.-CORREL.-DISJ.x}= =part{+COORD.+FOC.-DISJ} t{CORREL.x}  
 % *Pete said that either Jack likes apples or bananas (but I don't remember which)*.

Note that the approach to correlative coordinators like *either*, *neither*, *both* etc as focus particles initially attached to coordinator phrases before moving away from them is inspired by Zhang (2008), although owing to the constraints of the present EDMG formalism the focus particle must adjoin to the coordP, not to the coordinator head itself as Zhang proposes.

# Appendix B

## The grammar rules

This appendix contains all of the grammar rules from chapter 3 in one place for convenient reference.

### **B.1 Basic Merge and Move rules from section 3.3.2**

These rules have been reformulated to keep the head string of the head chain of an expression separate from its left and right dependents (to allow for head movement).

$$\frac{[\mathcal{E}, s_h, \mathcal{E} :: x = \gamma] \quad [t_l, t_h, t_r; x, \alpha_1, \dots, \alpha_k]}{[\mathcal{E}, s_h, t_l t_h t_r : \gamma, \alpha_1, \dots, \alpha_k]} (\text{merge1})$$

$$\frac{[t_l, t_h, t_r; x, \alpha_1, \dots, \alpha_k] \quad [\mathcal{E}, s_h, \mathcal{E} :: =x \gamma]}{[t_l t_h t_r, s, \mathcal{E} : \gamma, \alpha_1, \dots, \alpha_k]} (\text{merge2})$$

$$\frac{[t_l, t_h, t_r; x] \quad [s_l, s_h, s_r : =x \gamma, \alpha_1, \dots, \alpha_k]}{[t_l t_h t_r s_l, s_h, s_r : \gamma, \alpha_1, \dots, \alpha_k]} (\text{merge3})$$

$$\frac{[\mathcal{E}, s_h, \mathcal{E} :: x = \gamma] \quad [t_l, t_h, t_r; x \delta, \alpha_1, \dots, \alpha_k]}{[\mathcal{E}, s_h, \mathcal{E} : \gamma, t_l t_h t_r : \delta, \alpha_1, \dots, \alpha_k]} (\text{merge4})$$

$$\frac{[t_l, t_h, t_r; x \delta, \alpha_1, \dots, \alpha_k] \quad [\mathcal{E}, s_h, \mathcal{E} :: =x \gamma]}{[\mathcal{E}, s_h, \mathcal{E} : \gamma, t_l t_h t_r : \delta, \alpha_1, \dots, \alpha_k]} (\text{merge5})$$

$$\frac{[t_l, t_h, t_r; x \delta] \quad [s_l, s_h, s_r : =x \gamma, \alpha_1, \dots, \alpha_k]}{[s_l, s_h, s_r : \gamma, t_l t_h t_r : \delta, \alpha_1, \dots, \alpha_k]} (\text{merge6})$$

Figure B.1: The Merge rules from section 3.2.6 reformulated so that the head strings of head chains are kept separate from their left and right dependent strings in order to allow for head movement.

$$\frac{[s_l, s_h, s_r : +f \gamma, \alpha_1, \dots, \alpha_{i-1}, t : -f, \alpha_{i+1}, \dots, \alpha_k]}{[t s_l, s_h, s_r : \gamma, \alpha_1, \dots, \alpha_{i-1}, \alpha_{i+1}, \dots, \alpha_k]} (\text{move1})$$

$$\frac{[s_l, s_h, s_r : +f \gamma, \alpha_1, \dots, \alpha_{i-1}, t : -f \delta, \alpha_{i+1}, \dots, \alpha_k]}{[s_l, s_h, s_r : \gamma, \alpha_1, \dots, \alpha_{i-1}, t : \delta, \alpha_{i+1}, \dots, \alpha_k]} (\text{move2})$$

Figure B.2: The Move rules from section 3.2.6 reformulated so that the head strings of head chains are kept separate from their left and right dependent strings in order to allow for head movement.

## B.2 Adjunction rules from section 3.3.1

Again, these rules have been reformulated to keep the head string of an expression's head chain separate from its left and right dependents (to allow for head movement).

$$\frac{[t_l, t_h, t_r; x \approx] \quad [s_l, s_h, s_r; x \gamma, \alpha_1, \dots, \alpha_k]}{[t_l t_h t_r s_l, s_h, s_r; x \gamma, \alpha_1, \dots, \alpha_k]} (ad\ join1)$$

$$\frac{[s_l, s_h, s_r; x \gamma, \alpha_1, \dots, \alpha_k] \quad [t_l, t_h, t_r; \approx x]}{[s_l, s_h, s_r t_l t_h t_r; x \gamma, \alpha_1, \dots, \alpha_k]} (ad\ join2)$$

$$\frac{[t_l, t_h, t_r; x \approx \delta] \quad [s_l, s_h, s_r; x \gamma, \alpha_1, \dots, \alpha_k]}{[s_l, s_h, s_r; x \gamma, t_l t_h t_r; \delta, \alpha_1, \dots, \alpha_k]} (ad\ join3)$$

$$\frac{[s_l, s_h, s_r; x \gamma, \alpha_1, \dots, \alpha_k] \quad [t_l, t_h, t_r; \approx x \delta]}{[s_l, s_h, s_r; x \gamma, t_l t_h t_r; \delta, \alpha_1, \dots, \alpha_k]} (ad\ join4)$$

Figure B.3: Sub-functions of ADJOIN

### B.3 Head movement rules from section 3.3.2

$$\frac{[\varepsilon, s_h, \varepsilon:: >x = \gamma] \quad [t_l, t_h, t_r; x, \alpha_1, \dots, \alpha_k]}{[\varepsilon, t_h s_h, t_l t_r; \gamma, \alpha_1, \dots, \alpha_k]} (merge\_hm1)$$

$$\frac{[\varepsilon, s_h, \varepsilon:: >x = \gamma] \quad [t_l, t_h, t_r; x \delta, \alpha_1, \dots, \alpha_k]}{[\varepsilon, t_h s_h, \varepsilon; \gamma, t_l t_r; \delta, \alpha_1, \dots, \alpha_k]} (merge\_hm2)$$

$$\frac{[\varepsilon, s_h, \varepsilon:: x \leq \gamma] \quad [t_l, t_h, t_r; x, \alpha_1, \dots, \alpha_k]}{[\varepsilon, s_h t_h, t_l t_r; \gamma, \alpha_1, \dots, \alpha_k]} (merge\_hm3)$$

$$\frac{[\varepsilon, s_h, \varepsilon:: x \leq \gamma] \quad [t_l, t_h, t_r; x \delta, \alpha_1, \dots, \alpha_k]}{[\varepsilon, s_h t_h, \varepsilon; \gamma, t_l t_r; \delta, \alpha_1, \dots, \alpha_k]} (merge\_hm4)$$

Figure B.4: Head Movement functions

### B.4 Rightward Move rule from section 3.3.3

$$\frac{[s_l, s_h, s_r; x \gamma, \alpha_1, \dots, \alpha_{i-1}, t; x \sim, \alpha_{i+1}, \dots, \alpha_k]}{[s_l, s_h, s_r t; x \gamma, \alpha_1, \dots, \alpha_{i-1}, \alpha_{i+1}, \dots, \alpha_k]} r\_move$$

Figure B.5: The rule for rightward movement

## B.5 Covert Move rules from section 3.3.4

$$\frac{[\varepsilon, s_h, \varepsilon :: x = \gamma] \quad [t_l, t_h, t_r; x \delta, \alpha_1, \dots, \alpha_k]}{[\varepsilon, s_h, t_l t_h t_r : \gamma, \varepsilon : \delta, \alpha_1, \dots, \alpha_k]} (p\_merge1)$$

$$\frac{[t_l, t_h, t_r; x \delta, \alpha_1, \dots, \alpha_k] \quad [\varepsilon, s_h, \varepsilon :: = x \gamma]}{[t_l t_h t_r, s_h, \varepsilon : \gamma, \varepsilon : \delta, \alpha_1, \dots, \alpha_k]} (p\_merge2)$$

$$\frac{[t_l, t_h, t_r; x \delta] \quad [s_l, s_h, s_r : = x \gamma, \alpha_1, \dots, \alpha_k]}{[t_l t_h t_r, s_l, s_h, s_r : \gamma, \varepsilon : \delta, \alpha_1, \dots, \alpha_k]} (p\_merge3)$$

Figure B.6: Three phonetic Merge rules enabling subsequent covert movement.

$$\frac{[t_l, t_h, t_r; x \approx \delta] \quad [s_l, s_h, s_r; x \gamma, \alpha_1, \dots, \alpha_k]}{[s_l, s_h, s_r, t_l t_h t_r : x \gamma, \varepsilon : \delta, \alpha_1, \dots, \alpha_k]} (p\_adjoin1)$$

$$\frac{[s_l, s_h, s_r; x \gamma, \alpha_1, \dots, \alpha_k] \quad [t_l, t_h, t_r; \approx x \delta]}{[s_l, s_h, s_r, t_l t_h t_r : x \gamma, \varepsilon : \delta, \alpha_1, \dots, \alpha_k]} (p\_adjoin2)$$

Figure B.7: Phonetic Adjoin rules allowing the adjunct to subsequently undergo covert movement.

$$\frac{[s_l, s_h, s_r : +f \gamma, \alpha_1, \dots, \alpha_{i-1}, t : -f \delta, \alpha_{i+1}, \dots, \alpha_k]}{[t s_l, s_h, s_r : \gamma, \alpha_1, \dots, \alpha_{i-1}, \varepsilon : \delta, \alpha_{i+1}, \dots, \alpha_k]} (p\_move)$$

Figure B.8: The movement rule resulting in phonetic merge of the moving string and subsequent further covert movement of its formal features.

## B.6 Suicidal licensor rules from section 4.3.4

$$\frac{[s: +f!?\ \gamma, \alpha_1, \dots, \alpha_{i-1}, t: \delta, \alpha_{i+1}, \dots, \alpha_k]}{[s: \gamma, \alpha_1, \dots, \alpha_{i-1}, t: \delta, \alpha_{i+1}, \dots, \alpha_k]} (fcide)$$

$$\frac{[s: +f?\ \gamma, \alpha_1, \dots, \alpha_{i-1}, t: -f\ \delta, \alpha_{i+1}, \dots, \alpha_k]}{[s: \gamma, \alpha_1, \dots, \alpha_{i-1}, t: -f\ \delta, \alpha_{i+1}, \dots, \alpha_k]} (smove)$$

Figure B.9: Sub-functions of MOVE

## B.7 Control Move rules from section 3.3.6

$$\frac{[\varepsilon, s, \varepsilon :: d = \gamma] \quad [t_l, t_h, t_r; D \zeta, \alpha_1, \dots, \alpha_k]}{[\varepsilon, s, \varepsilon : \gamma, t_l t_h t_r : D \zeta, \alpha_1, \dots, \alpha_k]} (merge\_ctrl1)$$

$$\frac{[t_l, t_h, t_r; D \zeta, \alpha_1, \dots, \alpha_k] \quad [\varepsilon, s, \varepsilon :: =d \gamma]}{[\varepsilon, s, \varepsilon : \gamma, t_l t_h t_r : D \zeta, \alpha_1, \dots, \alpha_k]} (merge\_ctrl2)$$

$$\frac{[t_l, t_h, t_r; D \zeta] \quad [s_l, s_h, s_r :: =d \gamma, \alpha_1, \dots, \alpha_k]}{[s_l, s_h, s_r : \gamma, t_l t_h t_r : D \zeta, \alpha_1, \dots, \alpha_k]} (merge\_ctrl2)$$

Figure B.10: Merge rules with subsequent control movement

$$\frac{[s_l, s_h, s_r :: =d \gamma, \alpha_1, \dots, \alpha_{i-1}, t: D \zeta, \alpha_{i+1}, \dots, \alpha_k]}{[s_l, s_h, s_r : \gamma, \alpha_1, \dots, \alpha_{i-1}, t: D \zeta, \alpha_{i+1}, \dots, \alpha_k]} (move\_ctrl)$$

Figure B.11: Move rule in which D persists leading to further control movement



## B.8 Across-the-board phrasal movement rules from section 3.3.7.2

$$\frac{[t_l, t_h, t_r : x, \alpha_1^f, \dots, \alpha_l^f(\alpha_1^{s'}, \dots, \alpha_l^{s'})] \quad [s_l, s_h, s_r : =x\gamma, \alpha_1^f, \dots, \alpha_k^f(\alpha_1^s, \dots, \alpha_k^s)]}{[t_l t_h t_r s_l, s_h, s_r : \gamma, \alpha_1^f, \dots, \alpha_k^f(\alpha_1^s, \dots, \alpha_k^s)]} \quad (\text{merge} - \text{atb1})$$

$$\frac{[t_l, t_h, t_r : x\delta, \alpha_1^f, \dots, \alpha_l^f(\alpha_1^{s'}, \dots, \alpha_l^{s'})] \quad [s_l, s_h, s_r : =x\gamma, \alpha_1^f, \dots, \alpha_k^f(\alpha_1^s, \dots, \alpha_k^s)]}{[s_l, s_h, s_r : \gamma, t_l t_h t_r : \delta, \alpha_1^f, \dots, \alpha_k^f(\alpha_1^s, \dots, \alpha_k^s)]} \quad (\text{merge} - \text{atb2})$$

Figure B.12: Specifier Merge rules with ATB mover unification

$$\frac{[s_l, s_h, s_r : x\gamma, \alpha_1^f, \dots, \alpha_k^f(\alpha_1^s, \dots, \alpha_k^s)] \quad [t_l, t_h, t_r : \approx x, \alpha_1^f, \dots, \alpha_l^f(\alpha_1^{s'}, \dots, \alpha_l^{s'})]}{[s_l, s_h, s_r t_l t_h t_r : x\gamma, \alpha_1^f, \dots, \alpha_k^f(\alpha_1^s, \dots, \alpha_k^s)]} \quad (\text{ad join} - \text{atb1})$$

$$\frac{[s_l, s_h, s_r : x\gamma, \alpha_1^f, \dots, \alpha_k^f(\alpha_1^s, \dots, \alpha_k^s)] \quad [t_l, t_h, t_r : \approx x\delta, \alpha_1^f, \dots, \alpha_l^f(\alpha_1^{s'}, \dots, \alpha_l^{s'})]}{[s_l, s_h, s_r : x\gamma, t_l t_h t_r : \delta, \alpha_1^f, \dots, \alpha_k^f(\alpha_1^s, \dots, \alpha_k^s)]} \quad (\text{ad join} - \text{atb2})$$

Figure B.13: Right Adjoin rules with ATB mover unification

## B.9 Across-the-board head movement rules from section 3.3.7.4

$$\frac{[(\varepsilon, s_h, \varepsilon) \hat{=} \hat{x} = \hat{x}\gamma] \quad [(t_l, t_h, t_r) : x, \alpha_1, \dots, \alpha_k]}{[(\varepsilon, t_h, s_h t_l t_r) \hat{=} \hat{x}\gamma, \alpha_1, \dots, \alpha_k]} \quad (\text{mrg\_excorp})$$

$$\frac{[(t_l, t_h, t_r) : x, \alpha_1^f, \dots, \alpha_k^f(\alpha_1^{s'}, \dots, \alpha_k^{s'})] \quad [(s_l, s_h, s_r) \hat{=} \hat{x}\gamma, \alpha_1^f, \dots, \alpha_k^f(\alpha_1^s, \dots, \alpha_k^s)]}{[(t_l t_r s_l, s_h, s_r) \hat{=} \gamma, \alpha_1^f, \dots, \alpha_k^f(\alpha_1^s, \dots, \alpha_k^s)]} \quad (\text{mrg\_hm\_atb})$$

Figure B.14: Merge rules with excorporation and ATB head movement

## B.10 The rule allowing for list coordination from section 3.3.7.1

$$\frac{[t_l, t_h, t_r ; x] \quad [s_l, s_h, s_r \bar{\cdot} = x \gamma, \alpha_1, \dots, \alpha_k]}{[t_l t_h t_r s_l, s_h, s_r \bar{\cdot} = x \gamma, \alpha_1, \dots, \alpha_k]} (\text{merge3})$$

Figure B.15: The Merge rules for leftward conjuncts which allows the leftward selector to persist after checking in order to generate list coordination structures like *Tom, Dick and Harry*.

## B.11 Lexical head coordination rules from section 3.3.7.5

$$\frac{[s : \beta \cdot + f \gamma, \alpha_1, \dots, \alpha_{i-1}, t : \zeta \cdot - f \delta, \alpha_{i+1}, \dots, \alpha_k]}{[s : \beta + f \cdot \gamma, \alpha_1, \dots, \alpha_{i-1}, t : \zeta - f \cdot \delta, \alpha_{i+1}, \dots, \alpha_k]} (\text{move2} - \text{dot})$$

Figure B.16: Move2 with the dotted feature mechanism from Kobele (2008) added.

$$\frac{[\epsilon, s, \epsilon :: \cdot \beta x \gamma]}{[\epsilon, s, \epsilon :: \beta * x \gamma]} (\text{type-saturation})$$

$$\frac{[\epsilon, s, \epsilon \bar{\cdot} \cdot x = \bar{x} \beta x \gamma] \quad [\epsilon, t, \epsilon :: \beta * x \gamma]}{[\epsilon, s, t \bar{\cdot} x = \cdot = \bar{x} \beta x \gamma]} (\text{h\_coord1})$$

$$\frac{[\epsilon, t, \epsilon :: \beta * x \gamma] \quad [s_l, s_h, s_r \bar{\cdot} x = \cdot = \bar{x} \beta x \gamma]}{[t s_l, s_h, s_r \bar{\cdot} x = = \bar{x} \cdot \beta x \gamma]} (\text{h\_coord2})$$

$$\frac{[\epsilon, t, \epsilon :: \beta * x \gamma] \quad [s_l, s_h, s_r \bar{\cdot} x = \cdot = \bar{x} \beta x \gamma]}{[t s_l, s_h, s_r \bar{\cdot} x = \cdot = \bar{x} \beta x \gamma]} (\text{h\_coord3})$$

Figure B.17: Lexical head type-saturation and coordination rules

## B.12 Specifier escape rules from section 4.3.1.1

$$\frac{[t_l, t_h, t_r; x, \alpha_1^e, \dots, \alpha_k^e] \quad [s_l, s_h, s_r := x \gamma]}{[t_l t_h t_r s_l, s_h, s_r : \gamma, \alpha_1^e, \dots, \alpha_k^e]} (\text{merge\_esc\_1})$$

$$\frac{[t_l, t_h, t_r; x \delta, \alpha_1^e, \dots, \alpha_k^e] \quad [s_l, s_h, s_r := x \gamma]}{[s_l, s_h, s_r : \gamma, t_l t_h t_r : \delta, \alpha_1^e, \dots, \alpha_k^e]} (\text{merge\_esc\_2})$$

Figure B.18: Sub-functions of Merge allowing for escape from base generated specifier islands.

# Appendix C

## The syntactic and semantic PTB head finding rules

### C.1 Head-finding rules for PTB trees

This section gives the basic syntactic and semantic head finding rules used for extracting dependencies from PTB trees for the scoring of MG candidate trees during the automatic treebanking as described in section 5.4.4.1. The rules were adapted from the head-finding rules of Collins (1999). See section 5.4.4.1 for a description of how these rules work. The interested reader can find these rules in the file `autobank.py` and should also inspect the method `set_heads()` inside the class `Node` as there are various exception rules triggered there in the presence of certain function tags. For example, the tag `CC` is only chosen as a (syntactic or semantic) head child if it does not have any sisters marked with the `COORD` tag; if any `COORD` tags are present, then all constituents marked with this tag will be added as heads of that phrase instead of the `CC` constituent. Conversely, some function tags, such as `MNR`, `LOC`, `TMP` etc, prevent a constituent from being selected as a head child because they indicate that the item in question is a dependent.

#### C.1.1 Syntactic head-finding rules

**ADJP:** (JJ, L), (VBN, L), (NNS, L), (QP, L), ([NN, NNM, NNF, NNMF, NML], L), (\$, L), (ADVP, L), (VBG, L), ([ADJP, JJP], L), (JJR, L), (NP, L), (JJS, L), ([DT, DTSG, DTPL], L), (FW, L), (RBR, L), (RBS, L), (SBAR, L), (RB, L)

**JJP:** (NNS, L), (QP, L), ([NN, NNM, NNF, NNMF, NML], L), (\$, L), (ADVP, L), (JJ, L), (VBN, L), (VBG, L), ([ADJP, JJP], L), (JJR, L), (NP, L), (JJS, L), ([DT, DTSG, DTPL], L),

(FW, L), (RBR, L), (RBS, L), (SBAR, L), (RB, L)

**ADVP:** (RB, R), (RBR, R), (RBS, R), (FW, R), (ADVP, R), (TO, R), (CD, R), (JJR, R), (JJ, R), (IN, R), (NP, R), (JJS, R), ([NN, NNM, NNF, NNMF, NML], R)

**CONJP:** (CC, R), (RB, R), (IN, R)

**FRAG:** R

**INTJ:** L

**LST:** (LS, R), (:, R)

**NAC:** ([NN, NNM, NNF, NNMF, NML], L), (NNS, L), ([NNP, NNPM, NNPF, NNPMF], L), (NNPS, L), (NP, L), (NAC, L), (EX, L), (\$, L), (CD, L), (QP, L), ([PRP, PRP1SG, PRP2, PRP1PL, PRP3PL, PRPM3SG, PRPM3SG, PRPF3SG, PRPF3SG, PRP1SGSELF, PRP2SGSELF, PRP2PLELF, PRP1PLSELF, PRP3PLSELF, PRPM3SGSELF, PRPF3SGSELF, PRPM3SGSELF, PRPF3SGSELF], L), (VBG, L), (JJ, L), (JJS, L), (JJR, L), ([ADJP, JJP], L), (FW, L)

**NP:** (VP, R), ([DT, DTSG, DTPL], L), (QP, R), (NNS, R), ([NN, NNM, NNF, NNMF, NML], R), ([NNP, NNPM, NNPF, NNPMF], L), (NNPS, L), (NX, R), (NP, L)

**NX:** (VP, R), ([DT, DTSG, DTPL], L), (QP, R), (NNS, R), ([NN, NNM, NNF, NNMF, NML], R), ([NNP, NNPM, NNPF, NNPMF], L), (NNPS, R), (NX, R), (NP, L), (QP, R)

**NML:** (NNS, R), ([NN, NNM, NNF, NNMF, NML], R), ([NNP, NNPM, NNPF, NNPMF], L), (NNPS, L), (NX, R), (NP, L), (QP, R)

**PP:** (IN, R), (PP, R), (TO, R), (NP, R), ([NN, NNM, NNF, NNMF, NNS], R), ([NNP, NNPM, NNPF, NNPMF, NML], L), (S, R), (VBG, R), (VBN, R), (RP, R), (FW, R)

**PRN:** L

**PRT:** (RP, R)

**QP:** (CC, L), (\$, L), (IN, L), (NNS, L), ([NN, NNM, NNF, NNMF, NML], L), (JJ, L), (RB, L), ([DT, DTSG, DTPL], L), (CD, L), (NCD, L), (QP, L), (JJR, L), (JJS, L)

**RRC:** (VP, R), (NP, R), (ADVP, R), ([ADJP, JJP], R), (PP, R)

**S:** (IN, L), (S, L), (SBAR, L), (TO, L), (VP, L), ([ADJP, JJP], L), (UCP, L), (NP, R)

**SBAR:** (IN, L), (S, L), (SQ, L), (SINV, L), (SBAR, L), (FRAG, L)

**SBARQ:** (IN, L), (SQ, L), (S, L), (SINV, L), (SBARQ, L), (FRAG, L)

**SINV:** (VBZ, L), (VBD, L), (VBP, L), (VB, L), (MD, L), (VP, L), (S, L), (SINV, L), ([ADJP, JJP], L), (NP, L)

**SQ:** (VBZ, L), (VBD, L), (VBP, L), (VB, L), (MD, L), (VP, L), (SQ, L)

**UCP:** R

**VP:** (VBD, L), (TO, L), (VBN, L), (MD, L), (VBZ, L), (VB, L), (VBG, L), (VBP, L), (VP, L), ([ADJP, JJP], L), (:, L), ([NN, NNM, NNF, NNMF, NML], L), (NNS, L), (NP, L)

**WHADJP:** (CC, L), (WRB, L), (JJ, L), ([ADJP, JJP], L)

**WHADVP:** (CC, R), (WRB, R)

**WHNP:** (WDT, L), (WP, L), (WP\$, L), (WHADJP, L), (WHPP, L), (WHNP, L)

**WHPP:** (IN, R), (TO, R), (FW, R)

**BP:** (LRB, L)

### C.1.2 Semantic head-finding rules

**ADJP:** (JJ, L), (VBN, L), (NNS, L), (QP, L), ([NN, NNM, NNF, NNMF, NML], L), (\$, L), (ADVP, L), (VBG, L), ([ADJP, JJP], L), (JJR, L), (NP, L), (JJS, L), ([DT, DTSG, DTPL], L), (FW, L), (RBR, L), (RBS, L), (SBAR, L), (RB, L)

**JJP:** (NNS, L), (QP, L), ([NN, NNM, NNF, NNMF, NML], L), (\$, L), (ADVP, L), (JJ, L), (VBN, L), (VBG, L), ([ADJP, JJP], L), (JJR, L), (NP, L), (JJS, L), ([DT, DTSG, DTPL], L), (FW, L), (RBR, L), (RBS, L), (SBAR, L), (RB, L)

**ADVP:** (RB, R), (RBR, R), (RBS, R), (FW, R), (ADVP, R), (TO, R), (CD, R), (JJR, R), (JJ, R), (IN, R), (NP, R), (JJS, R), ([NN, NNM, NNF, NNMF, NML], R)

**CONJP:** (CC, R), (RB, R), (IN, R)

**FRAG:** R

**INTJ:** L

**LST:** (LS, R), (:, R)

**NAC:** ([NN, NNM, NNF, NNMF, NML], L), (NNS, L), ([NNP, NNPM, NNPF, NNPMF], L), (NNPS, L), (NP, L), (NAC, L), (EX, L), (\$, L), (CD, L), (QP, L), ([PRP, PRP1SG, PRP2, PRP1PL, PRP3PL, PRPM3SG, PRPM3SG, PRPF3SG, PRPF3SG, PRP1SG, PRP2SGSELF, PRP2SLSELF, PRP1PLSELF, PRP3PLSELF, PRPM3SGSELF, PRPF3SGSELF, PRPM3SGSELF, PRPF3SGSELF], L), (VBG, L), (JJ, L), (JJS, L), (JJR, L), ([ADJP, JJP], L), (FW, L)

**NP:** (VP, R), (NNS, R), ([NN, NNM, NNF, NNMF, NML], R), ([JJ, NNP, NNPM, NNPF, NNPMF, NNPS], L), (NNPS, L), (NX, R), (NP, L), (QP, R), ([DT, DTSG, DTPL], L)

**NX:** (VP, R), (NNS, R), ([NN, NNM, NNF, NNMF, NML], R), ([JJ, NNP, NNPM, NNPF, NNPMF, NNPS], L), (NNPS, L), (NX, R), (NP, L), (QP, R), ([DT, DTSG, DTPL], L)

**NML:** (NNS, R), ([NN, NNM, NNF, NNMF, NML], R), ([NNP, NNPM, NNPF, NNPMF], R), (NNPS, R), (NX, R), (NP, L), (QP, R)

**PP:** (PP, R), (NP, R), ([NN, NNM, NNF, NNMF, NNS, NNP, NNPM, NNPF, NNPMF, NML], R), (S, R), (TO, R), (VBG, R), (VBN, R), (IN, R), (RP, R), (FW, R)

**PRN:** L

**PRT:** (RP, R)

**QP:** (\$, L), (IN, L), (NNS, L), ([NN, NNM, NNF, NNMF, NML], L), (JJ, L), (RB, L), ([DT, DTSG, DTPL], L), (CD, L), (NCD, L), (QP, L), (JJR, L), (JJS, L)

**RRC:** (VP, R), (NP, R), (ADVP, R), ([ADJP, JJP], R), (PP, R)

**S:** (VP, L), (S, L), (SBAR, L), (TO, L), (IN, L), ([ADJP, JJP], L), (UCP, L), (NP, R)

**SBAR:** (S, L), (SQ, L), (SINV, L), (SBAR, L), (FRAG, L)

**SBARQ:** (SQ, L), (S, L), (SINV, L), (SBARQ, L), (FRAG, L)

**SINV:** (VP, L), (S, L), (SINV, L), (VBZ, L), (VBD, L), (VBP, L), (VB, L), (MD, L), ([ADJP, JJP], L), (NP, L)]

**SQ:** (SQ, L), (VP, L), (VBZ, L), (VBD, L), (VBP, L), (VB, L), (MD, L)

**UCP:** R

**VP:** (VP, L), (VBD, L), (TO, L), (VBN, L), (MD, L), (VBZ, L), (VB, L), (VBG, L), (VBP, L), ([ADJP, JJP], L), (:, L), ([NN, NNM, NNF, NNMF, NML], L), (NNS, L), (NP, L)

**WHADJP:** (CC, L), (WRB, L), (JJ, L), ([ADJP, JJP], L)

**WHADV:** (CC, R), (WRB, R)

**WHNP:** (NNS, R), ([NN, NNM, NNF, NNMF, NML], R), ([JJ, NNP, NNPM, NNPF, NNPMF, NNPS], L), (NNPS, L), (NX, R), (NP, L), (WDT, L), (WP, L), (WP\$, L), (WHADJP, L), (WHPP, L), (WHNP, L)

**WHPP:** (IN, R), (TO, R), (FW, R)

**BP:** (LRB, L)

# Appendix D

## The 100 sentences used for the global dependency evaluation of MGbank

### D.1 The gold test set for global evaluation of MGbank

The following is a list of the 100 sentences from the automatically generated section of MGbank which were hand-annotated after the treebank was generated and then used as a test set for the global evaluation of the automatically generated trees (see section 5.6.1).

wsj\_0022.mrg line: 14  
wsj\_0044.mrg line: 32  
wsj\_0044.mrg line: 42  
wsj\_0045.mrg line: 9  
wsj\_0051.mrg line: 21  
wsj\_0054.mrg line: 2  
wsj\_0098.mrg line: 9  
wsj\_0223.mrg line: 3  
wsj\_0231.mrg line: 21  
wsj\_0309.mrg line: 26  
wsj\_0319.mrg line: 8  
wsj\_0327.mrg line: 47  
wsj\_0331.mrg line: 9  
wsj\_0347.mrg line: 20  
wsj\_0405.mrg line: 33  
wsj\_0449.mrg line: 40



wsj\_0465.mrg line: 54  
wsj\_0465.mrg line: 88  
wsj\_0475.mrg line: 20  
wsj\_0524.mrg line: 8  
wsj\_0564.mrg line: 2  
wsj\_0564.mrg line: 40  
wsj\_0564.mrg line: 56  
wsj\_0587.mrg line: 12  
wsj\_0589.mrg line: 62  
wsj\_0589.mrg line: 20  
wsj\_0633.mrg line: 32  
wsj\_0723.mrg line: 5  
wsj\_0765.mrg line: 22  
wsj\_0766.mrg line: 39  
wsj\_0786.mrg line: 9  
wsj\_0808.mrg line: 8  
wsj\_0814.mrg line: 29  
wsj\_0946.mrg line: 22  
wsj\_0956.mrg line: 32  
wsj\_0956.mrg line: 40  
wsj\_0988.mrg line: 3  
wsj\_0999.mrg line: 44  
wsj\_1003.mrg line: 5  
wsj\_1006.mrg line: 17  
wsj\_1017.mrg line: 3  
wsj\_1038.mrg line: 7  
wsj\_1039.mrg line: 3  
wsj\_1047.mrg line: 16  
wsj\_1279.mrg line: 1  
wsj\_1350.mrg line: 2  
wsj\_1450.mrg line: 4  
wsj\_1453.mrg line: 32  
wsj\_1469.mrg line: 37  
wsj\_1495.mrg line: 13  
wsj\_1505.mrg line: 18

wsj\_1531.mrg line: 18  
wsj\_1547.mrg line: 22  
wsj\_1567.mrg line: 64  
wsj\_1590.mrg line: 5  
wsj\_1604.mrg line: 3  
wsj\_1616.mrg line: 20  
wsj\_1621.mrg line: 15  
wsj\_1621.mrg line: 5  
wsj\_1682.mrg line: 1  
wsj\_1682.mrg line: 20  
wsj\_1687.mrg line: 43  
wsj\_1688.mrg line: 5  
wsj\_1705.mrg line: 12  
wsj\_1709.mrg line: 1  
wsj\_1737.mrg line: 14  
wsj\_1748.mrg line: 8  
wsj\_1761.mrg line: 3  
wsj\_1774.mrg line: 11  
wsj\_1789.mrg line: 25  
wsj\_1792.mrg line: 21  
wsj\_1796.mrg line: 6  
wsj\_1811.mrg line: 1  
wsj\_1879.mrg line: 15  
wsj\_1903.mrg line: 12  
wsj\_1913.mrg line: 6  
wsj\_1936.mrg line: 49  
wsj\_1957.mrg line: 5  
wsj\_1982.mrg line: 4  
wsj\_2013.mrg line: 84  
wsj\_2045.mrg line: 16  
wsj\_2048.mrg line: 4  
wsj\_2054.mrg line: 5  
wsj\_2057.mrg line: 23  
wsj\_2100.mrg line: 43  
wsj\_2113.mrg line: 53

wsj\_2113.mrg line: 133

wsj\_2114.mrg line: 5

wsj\_2153.mrg line: 2

wsj\_2210.mrg line: 13

wsj\_2225.mrg line: 13

wsj\_2230.mrg line: 7

wsj\_2281.mrg line: 6

wsj\_2300.mrg line: 49

wsj\_2337.mrg line: 1

wsj\_2379.mrg line: 6

wsj\_2396.mrg line: 2

wsj\_2406.mrg line: 41

wsj\_2415.mrg line: 22

wsj\_2415.mrg line: 15

# Bibliography

- Abels, K. (2003). *Successive cyclicity, anti-locality, and adposition stranding*. PhD thesis, University of Connecticut Storrs, CT.
- Abels, K. (2004). Right node raising: Ellipsis or across the board movement. *Proceedings of NELS*, 34.
- Abels, K. (2007). Towards a restrictive theory of (remnant) movement: Improper movement, remnant movement, and a linear asymmetry. *Linguistic variation yearbook*, 7:53–120.
- Abney, S. (1996). Statistical methods and linguistics. In Klavans, J. and Resnik, P., editors, *The Balancing Act*, pages 1–26. MIT Press, Cambridge, MA.
- Abney, S. P. (1987). *The English noun phrase in its sentential aspect*. PhD thesis, Massachusetts Institute of Technology.
- Aboh, E. O. (1998). Focus constructions and the focus criterion in gungbe. *Linguistique Africaine*, (20):5–50.
- Adger, D. (2010). A minimalist theory of feature structure. *Features: Perspectives on a key notion in linguistics*, pages 185–218.
- Áfarli, T. A. (1994). A promotion analysis of restrictive relative clauses. *The Linguistic Review*, 11(2):81–100.
- Aldridge, E. (2009). Short wh-movement in old Japanese. *Japanese/Korean Linguistics*, 17:549–563.
- Baker, M. (2008). The macroparameter in a microparametric world. In Biberauer, T., editor, *The Limits of Syntactic Variation*, pages 351–374. John Benjamins.
- Baker, M. C. (1988). *Incorporation: A theory of grammatical function changing*. University of Chicago Press.

- Baltin, M. (1981). Strict bounding. In Baker, C. and McCarthy, J., editors, *The logical problem of language acquisition*, pages 257–295. MIT press, Cambridge, MA.
- Bangalore, S. and Joshi, A. (1999). Supertagging: An approach to almost parsing. *Computational Linguistics*, 25:237–265.
- Beil, F., Carroll, G., Prescher, D., Riezler, S., and Rooth, M. (1999). Inside-outside estimation of a lexicalized PCFG for German. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 269–276, College Park, Maryland, USA. Association for Computational Linguistics.
- Beil, F., Prescher, D., Schmid, H., and Schulte im Walde, S. (2002). Evaluation of the gramotron parser for german. In *Proceedings of the LREC Workshop: Beyond PARSEVAL, Las Palmas, Gran Canaria*, pages 52–59.
- Berwick, R. C. (1982). *Locality principles and the aquisition of syntactic knowledge*. PhD thesis, Massachusetts Inst. of Technology Cambridge.
- Berwick, R. C. (2015). Mind the gap. *Gallego, A., Ott, D.(eds.)*, 50:1–12.
- Berwick, R. C. and Epstein, S. D. (1995). Computational minimalism: The convergence of the minimalist syntactic program and categorial grammar. *TWLT-10: Algebraic Methods in Language Processing, Enschede, the Netherlands*.
- Bhatt, R. (2002). The raising analysis of relative clauses: Evidence from adjectival modification. *Natural language semantics*, 10(1):43–90.
- Bierwisch, M. (1963). *Grammatik des deutschen Verbs*. Akademie Verlag.
- Boeckx, C. (1999). Conflicting c-command requirements. *Studia Linguistica*, 53(3):227–250.
- Boeckx, C. (2009). *Understanding minimalist syntax: Lessons from locality in long-distance dependencies*. John Wiley Sons.
- Boeckx, C., Hornstein, N., and Nunes, J. (2010). *Control as Movement*. Cambridge University Press, Cambridge, UK.
- Boeckx, C. and Jeong, Y. (2004). The fine structure of intervention in syntax. *Issues in current linguistic theory: A festschrift for Hong Bae Lee*, pages 83–116.

- Bonato, R. and Retoré, C. (2001). Learning rigid lambek grammars and minimalist grammars from structured sentences. In *Third workshop on learning language in logic, Strasbourg*, pages 23–34.
- Borer, H. (1984). Parametric syntax: Case studies in semitic and romance languages, foris, dordrecht. *Borer Parametric Syntax: Case Studies in Semitic and Romance Languages 1984*.
- Borsley, R. (2005). Against conjp. *Lingua*, 115:461–482.
- Bos, J. (2008). Wide-coverage semantic analysis with boxer. In *Proceedings of the 2008 Conference on Semantics in Text Processing*, pages 277–286. Association for Computational Linguistics.
- Bošković, Ž. (2002). A-movement and the epp. *Syntax*, 5(3):167–218.
- Bošković, Ž. (2015). From the complex np constraint to everything: On deep extractions across categories. *The Linguistic Review*, 32(4):603–669.
- Bowers, J. (2002). Transitivity. *Linguistic inquiry*, 33(2):183–224.
- Brame, M. (1968). A new analysis of the relative clause: Evidence for an interpretive theory. *Unpublished manuscript, MIT*.
- Brants, S., Dipper, S., Hansen, S., Lezius, W., and Smith, G. (2002). The TIGER treebank. In *Proceedings of the workshop on treebanks and linguistic theories*, pages 24–41.
- Brennan, J. R., Stabler, E. P., Van Wagenen, S. E., Luh, W.-M., and Hale, J. T. (2016). Abstract linguistic structure correlates with temporal activity during naturalistic comprehension. *Brain and language*, 157:81–94.
- Bresnan, J. W. (1974). The position of certain clause-particles in phrase structure. *Linguistic Inquiry*, 5(4):614–619.
- Bruening, B. (2014). Defects of defective intervention. *Linguistic Inquiry*, 45(4):707–719.
- Burzio, L. (1986). *Italian syntax: A government-binding approach*, volume 1. Springer Science Business Media.
- Cann, R. (1999). Specifiers as secondary heads. *Specifiers, Minimalist Approaches*.

- Charniak, E. (2000). A maximum-entropy-inspired parser. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 132–139, Seattle, WA.
- Chen, J. (2001). *Towards efficient statistical parsing using lexicalized grammatical information*. PhD thesis, University of Delaware.
- Chen, J., Bangalore, S., and Vijay-Shanker, K. (2006). Automated extraction of tree-adjoining grammars from treebanks. *Natural Language Engineering*, 12:to appear.
- Chomsky, N. (1957). *Syntactic Structures*. Mouton, The Hague.
- Chomsky, N. (1964). *Current Issues in Linguistic Theory*. Mouton, The Hague.
- Chomsky, N. (1965). *Aspects of the Theory of Syntax*. MIT Press, Cambridge, MA.
- Chomsky, N. (1970). Remarks on nominalization. In Jacobs, R. and Rosenbaum, P., editors, *Readings in English Transformational Grammar*, pages 184–221. Ginn, Waltham, MA.
- Chomsky, N. (1973). Conditions on transformations. *A festschrift for Morris Halle*.
- Chomsky, N. (1977). On wh-movement. *Formal syntax*, pages 71–132.
- Chomsky, N. (1981). *Lectures on Government and Binding*. Foris, Dordrecht.
- Chomsky, N. (1986a). *Barriers*. MIT Press, Cambridge, MA.
- Chomsky, N. (1986b). *Knowledge of language: Its nature, origin, and use*. Greenwood Publishing Group.
- Chomsky, N. (1991). Some notes on economy of derivation and representation. *Anuario del Seminario de Filología Vasca "Julio de Urquijo"*, pages 53–82.
- Chomsky, N. (1992). Explaining language use. *Philosophical Topics*, 20(1):205–231.
- Chomsky, N. (1993). A minimalist program for linguistic theory. In Hale, K. and Keyser, S. J., editors, *The View from Building 20*, pages 1–52. MIT Press, Cambridge, MA. reprinted in Chomsky (1995).
- Chomsky, N. (1995). *The Minimalist Program*. MIT Press, Cambridge, Massachusetts.

- Chomsky, N. (2000). Minimalist inquiries: The framework. In Martin, R., Michaels, D., and Uriagereka, J., editors, *Step by Step: Essays in Minimalist Syntax in Honor of Howard Lasnik*, pages 89–155. MIT Press, Cambridge, MA.
- Chomsky, N. (2001a). Beyond explanatory adequacy. *MIT occasional papers in linguistics*, 20.
- Chomsky, N. (2001b). Derivation by phase. In Kenstowicz, M., editor, *Ken Hale: A life in language*, pages 1–52. MIT Press.
- Chomsky, N. (2004). The generative enterprise reconsidered. *Berlin: Mouton de Gruyter*.
- Chomsky, N. (2005). Three factors in language design. *Linguistic inquiry*, 36(1):1–22.
- Chomsky, N. (2008). On phases. *Current Studies in Linguistics Series*, 45:133.
- Chomsky, N. (2011). *The Generative Enterprise Revisited: Discussions with Riny Huybregts, Henk van Riemsdijk, Naoki Fukui and Mihoko Zushi*. Walter de Gruyter.
- Chomsky, N. (2013). Problems of projection. *Lingua*, 130:33–49.
- Chomsky, N. and Lasnik, H. (1977). Filters and control. *Linguistic inquiry*, 8(3):425–504.
- Chomsky, N. and Lasnik, H. (1993). The theory of principles and parameters. In Jacobs, J., von Stechow, A., Sternefeld, W., and Vennemann, T., editors, *Syntax: An international handbook of contemporary research*, volume 1, pages 506–569. Berlin: Walter de Gruyter.
- Cinque, G. (1999). *Adverbs and functional heads: A cross-linguistic perspective*. Oxford University Press on Demand.
- Cinque, G. (2006). *Restructuring and Functional Heads: The Cartography of Syntactic Structures Volume 4*, volume 4. Oxford University Press on Demand.
- Clark, B. (2000). Some things are not susceptible to thinking about: The historical development of tough complementation. ms.
- Clark, S. and Curran, J. (2007a). Formalism-independent parser evaluation with CCG and depbank. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 248–255.



- Clark, S. and Curran, J. R. (2004). The importance of supertagging for wide-coverage CCG parsing. In *Proceedings of the 20th international conference on Computational Linguistics*, page 282. Association for Computational Linguistics.
- Clark, S. and Curran, J. R. (2007b). Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics*, 33:493–552.
- Clark, S. and Curran, J. R. (2010). Supertagging for efficient wide-coverage CCG parsing. In Bangalore, S. and Joshi, A. K., editors, *Supertagging: Using Complex Lexical Descriptions in Natural Language Processing*, pages 221–233. MIT Press.
- Clark, S. and Hockenmaier, J. (2002). Evaluating a wide-coverage CCG parser. In *Proceedings of the LREC 2002 "Beyond Parseval" Workshop*, pages 60–66, Las Palmas.
- Clark, S., Steedman, M., and Curran, J. R. (2004). Object-extraction and question-parsing using CCG. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*.
- Collins, C. (2002). Eliminating labels. *Derivation and explanation in the Minimalist Program*, pages 42–64.
- Collins, C. (2005). A smuggling approach to the passive in english. *Syntax*, 8(2):81–120.
- Collins, M. (1999). *Head-Driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania.
- Cormack, A. and Smith, N. (2000). Head movement and negation in english. *Transactions of the Philological Society*, 98(1):49–85.
- Cormack, A. and Smith, N. (2002). Compositionality, copy theory, and control. *University College of London Working Papers in Linguistics*, 14:355–373.
- Cramer, B. and Zhang, Y. (2009). Construction of a german HPSG grammar from a detailed treebank. In *Proceedings of the GEAF workshop ACL- IJCNLP 2009*, pages 37–45.
- Cramer, B. and Zhang, Y. (2010). Constraining robust constructions for broad-coverage parsing with precision grammars. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 223–231, Beijing.

- Demberg, V. and Keller, F. (2008). A psycholinguistically motivated version of tag. In *Proceedings of the Ninth International Workshop on Tree Adjoining Grammar and Related Frameworks (TAG+ 9)*, pages 25–32.
- deMena Travis, L. (1984). *Parameters and Effects of Word Order Variation*. PhD thesis, MIT, Cambridge, Massachusetts.
- Den Dikken, M. (1995). Binding, expletives, and levels. *Linguistic inquiry*, 26(2):347–354.
- Dixon, R. M., Dixon, R. M. W., and Vogel, A. R. (2004). *The Jarawara language of southern Amazonia*. Oxford University Press on Demand.
- Dorr, B. J. (1991). Principle-based parsing for machine translation. In *Principle-Based Parsing*, pages 153–183. Springer.
- Dozat, T. and Manning, C. D. (2017). Deep bi-affine attention for neural dependency parsing. In *International Conference on Learning Representations*.
- Dyer, C., Ballesteros, M., Ling, W., Matthews, A., and Smith, N. A. (2015). Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 334–343. Association for Computational Linguistics.
- Dyer, C., Kuncoro, A., Ballesteros, M., and Smith, N. A. (2016). Recurrent neural network grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 199–209, San Diego, California. Association for Computational Linguistics.
- Emonds, J. (1978). The verbal complex V'-V in french. *Linguistic inquiry*, pages 151–175.
- Epstein, S. and Seely, D. (2006). *Derivations In Minimalism*. Cambridge University Press, Cambridge.
- Epstein, S. D., Kitahara, H., and Seely, T. D. (2015). *Explorations in maximizing syntactic minimization*. Routledge.

- Ermolaeva, M. (2017). Morphological agreement in minimalist grammars. In *International Conference on Formal Grammar*, pages 20–36. Springer.
- Ernst, T. (2002). *The Syntax of Adjuncts*. Cambridge University Press, Cambridge, UK.
- Felser, C. (2004). Wh-copying, phases, and successive cyclicity. *Lingua*, 114(5):543–574.
- Ficler, J. and Goldberg, Y. (2016). Coordination annotation extension in the penn tree bank. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, Berlin, German, Volume 1: Long Papers*, pages 834–842.
- Flickinger, D., Zhang, Y., and Kordoni, V. (2012). Deepbank. a dynamically annotated treebank of the wall street journal. In *Proceedings of the 11th International Workshop on Treebanks and Linguistic Theories*, pages 85–96.
- Fong, S. (1991). The computational implementation of principle-based parsers. In Berwick, R., Abney, S., and Tenny, C., editors, *Principle-Based Parsing*, pages 65–82. Kluwer, Dordrecht.
- Fong, S. and Ginsburg, J. (2012). Computation with doubling constituents: Pronouns and antecedents in phase theory. *AMD Sciullo, Hg., Towards a Biolinguistic understanding of Grammar: Essays on Interfaces. Amsterdam/Philadelphia: John Benjamins Publishing Co*, pages 303–338.
- Fowlie, M. (2013). Order and optionality: Minimalist grammars with adjunction. In *Proceedings of the 13th Meeting on the Mathematics of Language (MoL 13)*, pages 12–20.
- Fowlie, M. (2015). *Minimalist Grammars with Adjunction*. PhD thesis, UCLA.
- Fowlie, M. and Koller, A. (2017). Parsing minimalist languages with interpreted regular tree grammars. In *Proceedings of the Thirteenth International Workshop on Tree Adjoining Grammar and Related Formalisms (TAG+13)*, pages 11–20.
- Fox, D. (2000). *Economy and Semantic Interpretation*. MIT Press, Cambridge, MA.
- Frampton, J. and Gutmann, S. (2000). Agreement is feature sharing. *Ms., Northeastern University*.

- Frey, W. and Gärtner, H.-M. (2002). On the treatment of scrambling and adjunction in minimalist grammars. In Jäger, G., Monachesi, P., Penn, G., and Wintner, S., editors, *Proceedings of Formal Grammar*, pages 41–52.
- Friedman, J. (1971). *A Computer Model of Transformational Grammar*. Elsevier, New York.
- Gärtner, H.-M. and Michaelis, J. (2003). A note on countercyclicity and minimalist grammars. In *Proceedings of FGVienna: The 8th Conference on Formal Grammar*, pages 95–109.
- Gärtner, H.-M. and Michaelis, J. (2005). A note on the complexity of constraint interaction: Locality conditions and minimalist grammars. In *International Conference on Logical Aspects of Computational Linguistics*, pages 114–130. Springer.
- Gärtner, H.-M. and Michaelis, J. (2007). Some remarks on locality conditions and minimalist grammars. *Interfaces+ recursion= language*, 89:161–195.
- Gärtner, H.-M. and Michaelis, J. (2010). On the treatment of multiple-wh-interrogatives in minimalist grammars. In *Language and Logos*, volume 72, pages 339–336. Akademie Verlag, Berlin.
- Gazdar, G. (1981). Unbounded dependencies and coordinate structure. In *The Formal complexity of natural language*, pages 183–226. Springer.
- Gazdar, G., Klein, E. H., Pullum, G. K., and Sag, I. A. (1985). *Generalized Phrase Structure Grammar*. Harvard University Press, Cambridge, Massachusetts.
- Gerth, S. (2015). *Memory limitations in sentence comprehension*. PhD thesis, University of Potsdam.
- Graf, T. (2018). Grammar size and quantitative restrictions on movement. In *Proceedings of the Society for Computation in Linguistics*, pages 23–33.
- Graf, T., Aksënova, A., and De Santo, A. (2015a). A single movement normal form for minimalist grammars. In *Formal Grammar*, pages 200–215. Springer.
- Graf, T., Fodor, B., Monette, J., Rachiele, G., Warren, A., and Zhang, C. (2015b). A refined notion of memory usage for minimalist parsing. In *Proceedings of the 14th Meeting on the Mathematics of Language (MoL 2015)*, pages 1–14.

- Graf, T. and Marcinek, B. (2014). Evaluating evaluation metrics for minimalist parsing. In *Proceedings of the Fifth Workshop on Cognitive Modeling and Computational Linguistics*, pages 28–36.
- Graf, T., Monette, J., and Zhang, C. (2017). Relative clauses as a benchmark for minimalist parsing. *Journal of Language Modelling*, 5(1):57–106.
- Graves, A. (2013). Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.
- Grimshaw, J. (1991). Extended projection. Unpublished manuscript, Brandeis University, Waltham, Mass. (Also appeared in J. Grimshaw (2005), *Words and Structure*, Stanford: CSLI).
- Grohmann, K. K. (2000). *Prolific peripheries: A radical view from the left*. PhD thesis, University of Maryland, College Park.
- Gross, M. (1978). On the failure of generative grammar. *Language*, 55:859–885.
- Haegeman, L. (2012). *Adverbial clauses, main clause phenomena, and composition of the left periphery: The cartography of syntactic structures*, volume 8. Oxford University Press.
- Hale, J. (2001). A probabilistic early parser as a psycholinguistic model. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–8. Association for Computational Linguistics.
- Hale, J. (2003). *Grammar, Uncertainty and Sentence Processing*. PhD thesis, Johns Hopkins University.
- Hale, J., Dyer, C., Kuncoro, A., and Brennan, J. (2018). Finding syntax in human encephalography with beam search. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2727–2736, Melbourne, Australia. Association for Computational Linguistics.
- Hale, J. T. (2011). What a rational parser would do. *Cognitive Science*, 35(3):399–443.
- Harkema, H. (2001). *Parsing Minimalist Languages*. PhD thesis, UCLA, Los Angeles, California.

- Hartman, J. (2011). (non-) intervention in a-movement: Some cross-constructural and cross-linguistic considerations. *Linguistic Variation*, 11(2):121–148.
- Hartmann, K. (2001). *Right node raising and gapping: Interface conditions on prosodic deletion*. John Benjamins Publishing.
- Hauser, M. D., Chomsky, N., and Fitch, W. T. (2002). The faculty of language: What is it, who has it, and how did it evolve? *science*, 298(5598):1569–1579.
- Henry, A. and Cottell, S. (2007). A new approach to transitive expletives: evidence from belfast english. *English Language Linguistics*, 11(2):279–299.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Hockenmaier, J. and Steedman, M. (2002a). Acquiring compact lexicalized grammars from a cleaner treebank. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC)*, pages 1974–1981.
- Hockenmaier, J. and Steedman, M. (2002b). Generative models for statistical parsing with combinatory categorial grammar. In *In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 335–342.
- Hockenmaier, J. and Steedman, M. (2007). CCGbank: A corpus of CCG derivations and dependency structures extracted from the penn treebank. *Computational Linguistics*, 33(3):355–396.
- Holmberg, A. and Roberts, I. (2013). The syntax–morphology relation. *Lingua*, 130:111–131.
- Hornstein, N. (2001). *Move! A Minimalist Theory of Construal*. Blackwell Publishing.
- Hornstein, N. (2009). A theory of syntax. *Cambridge: CUP*.
- Hornstein, N. and Kiguchi, H. (2003). Pro gate and movement. *University of Pennsylvania Working Papers in Linguistics*, 8(1):9.
- Hornstein, N., Nunes, J., and Grohmann, K. (2005). *Understanding Minimalism*. Cambridge University Press.
- Huang, C.-T. J. (1982). *Logical relations in Chinese and the theory of grammar*. PhD thesis, MIT, Cambridge, Massachusetts.

- Hudson, R. A. (1976). Conjunction reduction, gapping, and right-node raising. *Language*, pages 535–562.
- Hunter, T. (2018). Left-corner parsing of minimalist grammars. Under submission.
- Hunter, T. and Dyer, C. (2013). Distributions on minimalist grammar derivations. In *Proceedings of the 13th Meeting on the Mathematics of Language (MoL 13)*, pages 1–11, Sofia, Bulgaria. The Association of Computational Linguistics.
- Iatridou, S., Anagnostopoulou, E., and Izvorski, R. (2001). Observations about the form and meaning of the perfect. In: *Kenstowicz, M. (Ed.), Ken Hale: A Life in Language*, pages 189–238.
- im Walde, S. S., Schmid, H., Rooth, M., Riezler, S., and Prescher, D. (2001). Statistical grammar models and lexicon acquisition. In *Linguistic Form and its Computation*, pages 59–110. CSLI, Stanford, CA.
- Jackendoff, R. (1977). *X-bar syntax*. The MIT Press.
- Jacobson, P. (1987). Review of gerald gazdar, ewan klein, geoffrey k. pullum, and ivan a. sag. generalized phrase structure grammar. *Linguistics and Philosophy*, 10:389–426.
- Jacobson, P. (1992). The lexical entailment theory of control and the tough construction. In Sag, I. and Szabolcsi, A., editors, *Lexical Matters*, pages 269–300. CSLI Publications, Stanford, CA.
- Johnson, K. (1991a). Object positions. *Natural Language Linguistic Theory*, 9(4):577–636.
- Johnson, M. (1991b). Deductive parsing: The use of knowledge of language. In *Principle-Based Parsing*, pages 39–64. Springer.
- Joshi, A. (1985). Tree-adjoining grammars. In Dowty, D., Karttunen, L., and Zwicky, A., editors, *Natural Language Parsing*, pages 206–250. Cambridge University Press, Cambridge.
- Kanazawa, M., Michaelis, J., Salvati, S., and Yoshinaka, R. (2011). Well-nestedness properly subsumes strict derivational minimalism. In *International Conference on Logical Aspects of Computational Linguistics*, pages 112–128. Springer.

- Kasai, J., Frank, R., McCoy, R. T., Rambow, O., and Nasr, A. (2017). Improved CCG parsing with semi-supervised supertagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1713–1723.
- Kayne, R. (2002). Pronouns and their antecedents. In Epstein, S. and Seely, D., editors, *Derivation and Explanation in the Minimalist Program*, pages 133–166. Blackwell Publishing.
- Kayne, R. S. (1984). *Connectedness and binary branching*. Foris, Dordrecht.
- Kayne, R. S. (1994). *The Antisymmetry of Syntax, Linguistic Inquiry Monograph Twenty-Five*. MIT Press, Cambridge, Massachusetts.
- Keenan, E. L. and Hawkins, S. (1987). The psychological validity of the accessibility hierarchy. *Universal grammar*, 15:60–85.
- Kiguchi, H. (2002). *Syntax unchained*. PhD thesis, University of Maryland.
- Kim, B. (1995). Predication in tough constructions. In *Proceedings of the fourteenth West Coast conference on formal linguistics*, pages 271–285.
- King, M. (1983). Transformational parsing. *Parsing Natural Language*, pages 19–34.
- Kitahara, H. (1997). *Elementary operations and optimal derivations*. Number 31 in Linguistic Inquiry Monograph 31. MIT Press.
- Klein, D. and Manning, C. D. (2003). A\* parsing: fast exact viterbi parse selection. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 40–47. Association for Computational Linguistics.
- Kobele, G. M. (2005). Features moving madly: A formal perspective on feature percolation in the minimalist program. *Research on Language and Computation*, 3(4):391–410.
- Kobele, G. M. (2006). *Generating Copies: An investigation into structural identity in language and grammar*. PhD thesis, University of California, Los Angeles.
- Kobele, G. M. (2008). Across-the-Board Extraction in Minimalist Grammars. In *Proceedings of the Ninth International Workshop on Tree Adjoining Grammar and Related Formalisms (TAG+9)*, volume 9, pages 113–128.



- Kobele, G. M. (2010). Without remnant movement, MGs are context-free. In *The Mathematics of Language*, pages 160–173. Springer.
- Kobele, G. M., Gerth, S., and Hale, J. (2013). Memory resource allocation in top-down minimalist parsing. In *Formal Grammar*, pages 32–51. Springer.
- Kobele, G. M. and Michaelis, J. (2009). Two type 0-variants of minimalist grammars. 25, pages 81–91.
- Kobele, G. M. and Michaelis, J. (2011). Disentangling notions of specifier impenetrability: Late adjunction, islands, and expressive power. In *The mathematics of language*, pages 126–142. Springer.
- Koizumi, M. (1995). *Phrase structure in minimalist syntax*. PhD thesis, Massachusetts Inst. of Technology Cambridge.
- Koopman, H. and Sportiche, D. (1991). The position of subjects. *Lingua*, 85(2-3):211–258.
- Koopman, H. J. and Szabolcsi, A. (2000). *Verbal complexes*. Number 34 in Current studies in linguistics 34. MIT Press, Cambridge, MA.
- Kuhns, R. J. (1990). A PARLOG implementation of government-binding theory. In *13th International Conference on Computational Linguistics, COLING 1990, University of Helsinki, Finland, August 20-25, 1990*, pages 394–396.
- Larson, R. (1988). On the double object construction. *Linguistic Inquiry*, 19:335–392.
- Larson, R. K. (2009). *Grammar as science*. MIT Press.
- Larson, R. K., Warren, D. S., Freie, J., and Sagonas, K. (1996). *SYNTACTICA. Version 1.0 for Nextstep. Incl. 1 CD*. London: MIT Press.
- Lasnik, H. (1997). Levels of representation and the elements of anaphora. *Atomism and binding*, 1:251.
- Lasnik, H. and Fiengo, R. (1974). Complement object deletion. *Linguistic Inquiry*, 5(4):535–571.
- Lasnik, H. and Saito, M. (1991). On the subject of infinitives. In *27th regional meeting of the Chicago Linguistic Society*, volume 27, pages 324–343.

- Lebeaux, D. (1985). Locality and anaphoric binding. *The Linguistic Review*, 4(4):343–364.
- Lebeaux, D. (1988). *Language acquisition and the form of grammar*. PhD thesis, University of Massachusetts.
- Legate, J. A. (2002). Phases in “beyond explanatory adequacy”. *Manuscript, MIT, Cambridge*.
- Levine, R. D. (1985). Right node (non-) raising. *Linguistic Inquiry*, 16(3):492–497.
- Levine, R. D. (2001). The extraction riddle: just what are we missing? *Journal of Linguistics*, 37(1):145–174.
- Lewis, M., Lee, K., and Zettlemoyer, L. (2016). LSTM CCG parsing. In *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 221–231.
- Lewis, M. and Steedman, M. (2014a). A\* CCG parsing with a supertag-factored model. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 990–1000.
- Lewis, M. and Steedman, M. (2014b). Improved CCG parsing with semi-supervised supertagging. *Transactions of the Association for Computational Linguistics*, 2(1):327–338.
- Li, M. and Vitányi, P. M. (1992). Inductive reasoning and kolmogorov complexity. *Journal of Computer and System Sciences*, 44(2):343–384.
- Lidz, J. and Idsardi, W. (1998). Chains and phono-logical form. *U. Penn Working Papers in Linguistics*, 5:109–125.
- Lightfoot, D. (1976). The base component as a locus of syntactic change. current progress in historical linguistics, ed. by william christie jr., 17-37.
- Lin, D. (1993). Principle-based parsing without overgeneration. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, pages 112–120. Association for Computational Linguistics.
- Lin, D. (1998). Dependency-based evaluation of minipar. In *Proceedings of the Workshop on the Evaluation of Parsing Systems, First International Conference on Language Resources and Evaluation*, Granada, Spain.

- Lin, D. (2001). Latat: Language and text analysis tools. In *Proceedings of the first international conference on Human language technology research*, pages 1–6. Association for Computational Linguistics.
- Magerman, D. M. (1994). *Natural Language Parsing As Statistical Pattern Recognition*. PhD thesis, Stanford University, Stanford, CA, USA. UMI Order No. GAX94-22102.
- Mainguy, T. (2010). A probabilistic top-down parser for minimalist grammars. *arXiv preprint arXiv:1010.1826*.
- Maling, J. M. (1972). On “gapping and the order of constituents”. *Linguistic Inquiry*, pages 101–108.
- Manetta, E. (2010). Wh-expletives in hindi-urdu: the vp phase. *Linguistic Inquiry*, 41(1):1–34.
- Manning, C. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA.
- Marcus, M., Santorini, B., and Marcinkiewicz, M. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19:313–330.
- Marcus, M. P. (1980). *Theory of syntactic recognition for natural languages*. MIT press.
- Martin, R. (2001). Null case and the distribution of pro. *Linguistic inquiry*, 32(1):141–166.
- Martin, R. A. (1996). *A minimalist theory of PRO and control*. PhD thesis, University of Connecticut.
- Matushansky, O. (2005). Going through a phase. *MIT working papers in linguistics*, 49:157–181.
- McCloskey, J. (1999). On the right edge in irish. *Syntax*, 2(3):189–209.
- Meyers, A., Reeves, R., Macleod, C., Szekely, R., Zielinska, V., Young, B., and Grishman, R. (2004). The NomBank project: An interim report. In *Proceedings of HLT-EACL Workshop: Frontiers in Corpus Annotation*.

- Michaelis, J. (1998). Derivational minimalism is mildly context-sensitive. In *International Conference on Logical Aspects of Computational Linguistics*, pages 179–198. Springer.
- Michaelis, J. (2004). Observations on strict derivational minimalism. *Electronic Notes in Theoretical Computer Science*, 53:192–209.
- Michaelis, J. and Kobele, G. (2005). Two type 0-variants of minimalist grammars. In *The 10th conference on Formal Grammar and The 9th Meeting on Mathematics of Language*.
- Mikkelsen, L. (2005). *Copular clauses: Specification, predication and equation*, volume 85. John Benjamins Publishing.
- Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. J. (1990). Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244.
- Miyao, Y. (2006). From linguistic theory to syntactic analysis: Corpus-oriented grammar development and feature forest model. *PhD thesis, University of Tokyo*.
- Miyao, Y., Ninomiya, T., and Tsujii, J. (2005). Corpus-oriented grammar development for acquiring a head-driven phrase structure grammar from the penn treebank. In *Proceedings of the First International Joint Conference on Natural Language Processing, IJCNLP'04*, pages 684–693, Berlin, Heidelberg. Springer-Verlag.
- Monahan, P. J. (2003). *Backward object control in Korean*. PhD thesis, University of Florida.
- Montague, R. (1970). English as a formal language. In Visentini, B., editor, *Linguaggi nella Società e nella Technica*, pages 189–224. Edizioni di Comunità, Milan.
- Müller, S. (2016). *Grammatical theory: From transformational grammar to constraint-based approaches*. Language Science Press.
- Neumann, G. (1998). Automatic extraction of stochastic lexicalized tree grammars from treebanks. In *Proceedings of the 4th Workshop on Tree-Adjoining Grammars and Related Frameworks (TAG+4), August 1-3, Philadelphia, USA*.

- Nivre, J., Rimell, L., McDonald, R., and Gomez-Rodriguez, C. (2010). Evaluation of dependency parsers on unbounded dependencies. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 833–841. Association for Computational Linguistics.
- Nunes, J. (1995). *The Copy Theory of Movement and Linearisation of Chains in the Minimalist Program*. PhD thesis, University of Maryland, College Park.
- Nunes, J. (2001). Sideward movement. *Linguistic Inquiry*, 32:303–344.
- Nunes, J. (2004). *Linearization of Chains and Sideward Movement*. MIT Press, Cambridge, Massachusetts.
- Oepen, S. (2001). [incr tsdb()] — competence and performance laboratory. User manual. Technical report, Computational Linguistics, Saarland University, Saarbrücken, Germany. in preparation.
- Palmer, M., Gildea, D., and Kingsbury, P. (2005). The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.
- Partee, B. (1975). Montague grammar and transformational grammar. *Linguistic inquiry*, pages 203–300.
- Partee, B. and Rooth, M. (1983). Generalized conjunction and type ambiguity. In *Meaning, Use, and Interpretation of Language*. Berlin: de Gruyter.
- Perlmutter, D. (1968). *Deep and surface structure constraints in syntax*. PhD thesis, Massachusetts Institute of Technology.
- Pesetsky, D. (2015). complementizer-trace effects. Unpublished MS., MIT.
- Pesetsky, D. M. (1996). *Zero Syntax: Experiencers and Cascades*. Number 27 in Current studies in linguistics 27. MIT press.
- Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proc. of NAACL*.
- Peters, P. S. and Ritchie, R. W. (1973). On the generative power of transformational grammars. *Information sciences*, 6:49–83.
- Petrick, S. R. (1965). *A recognition procedure for transformational grammars*. PhD thesis, Massachusetts Institute of Technology.

- Pires, A. (2001). Clausal and tp-defective gerunds: Control without tense. In *PROCEEDINGS-NELS*, volume 31, pages 389–406.
- Polinsky, M. and Potsdam, E. (2002). Backward control. *Linguistic Inquiry*, 33(2):245–282.
- Pollard, C. and Sag, I. (1994). *Head Driven Phrase Structure Grammar*. CSLI Publications, Stanford, CA.
- Pollock, J.-Y. (1989). Verb movement, universal grammar, and the structure of ip. *Linguistic Inquiry*, 20:365–424.
- Portelance, E., Bergen, L., Bruno, C., and O’Donnell, T. J. (2017). Mildly context sensitive grammar induction and variational bayesian inference. *CoRR*, abs/1710.11350.
- Postal, P. (1974). *On Raising: One Rule of English Grammar and its Theoretical Implications*. MIT Press.
- Postal, P. (1998). *Three Investigations of Extraction*. MIT Press, Cambridge, MA.
- Pullum, G. (2009). Computational linguistics and generative linguistics: The triumph of hope over experience. In *Proceedings of the EACL 2009 Workshop on the Interaction between Linguistics and Computational Linguistics*, pages 12–21, Athens, Greece. Lawrence Erlbaum Associates.
- Pullum, G. K. and Rawlins, K. (2007). Argument or no argument? *Linguistics and Philosophy*, 30(2):277–287.
- Quine, W. V. O. (1960). *Word and Object*. MIT Press, Cambridge, MA.
- Radford, A. (1988). *Transformational grammar: A first course*, volume 1. Cambridge University Press.
- Radford, A. (2004). *Minimalist Syntax: Exploring the Structure of English*. Cambridge University Press.
- Reinhart, T. and Reuland, E. (1991). 14 anaphors and logophors: an argument structure perspective. *Long distance anaphora*, page 283.
- Reinhart, T. and Reuland, E. (1993). Reflexivity. *Linguistic inquiry*, 24(4):657–720.

- Richards, N. W. (1997). *What moves where when in which languages?* PhD thesis, Massachusetts Institute of Technology.
- Rimell, L., Clark, S., and Steedman, M. (2009). Unbounded dependency recovery for parser evaluation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 813–821, Singapore. ACL.
- Rissanen, J. and Ristad, E. S. (1994). Language acquisition in the mdl framework. *Language computations*, pages 149–166.
- Rizzi, L. (1990). *Relativized minimality*. The MIT Press.
- Rizzi, L. (1997). The fine structure of the left periphery. In *Elements of grammar*, pages 281–337. Springer.
- Rizzi, L. (2004). Locality and left periphery. *Structures and beyond: The cartography of syntactic structures*, 3:223–251.
- Rizzi, L. (2006). On the form of chains: Criterial positions and ecp effects. *Current Studies in Linguistics Series*, 42:97.
- Roberts, I. G. (2010). *Agreement and head movement: Clitics, incorporation, and defective goals*, volume 59. MIT Press.
- Rosén, V., Meurer, P., and De Smedt, K. (2009). LFG parsebanker: A toolkit for building and searching a treebank as a parsed corpus. In *Proceedings of the seventh international workshop on treebanks and linguistic theories (TLT7)*, pages 127–133.
- Ross, J. R. (1967). *Constraints on variables in syntax*. PhD thesis, MIT, Cambridge, Massachusetts.
- Sabbagh, J. (2003). Ordering and linearizing rightward movement. *Proceedings of the 22nd WCCFL*, 436, 449.
- Sabel, J. (2000). *Expletives as features*. Univ.-Bibliothek Frankfurt am Main.
- Saddy, D. (1991). Wh-scope mechanisms in bahasa indonesia. *MIT working papers in linguistics*, 15:183–218.
- Salvati, S. (2011). Minimalist grammars in the light of logic. In *Logic and grammar*, pages 81–117. Springer.

- Sauerland, U. (1999). Erasability and interpretation. *Syntax*, 2(3):161–188.
- Schachter, P. (1973). Focus and relativization. *Language*, pages 19–46.
- Schmid, H. (2000). *YAP: Parsing and Disambiguation With Feature-Based Grammars*. PhD thesis, Institute for Computational Linguistics, University of Stuttgart, Germany.
- Seki, H., Matsumura, T., Fujii, M., and Kasami, T. (1991). On multiple context free grammars. *Theoretical Computer Science*, 88:191–229.
- Sheehan, M. (2011). A note on case-assignment to cp. *Snippets*, 24:18–19.
- Sportiche, D. (1988). A theory of floating quantifiers and its corollaries for constituent structure. *Linguistic inquiry*, 19(3):425–449.
- Stabler, E. P. (1984). Berwick and Weinberg on linguistics and computational psychology. *Cognition*, 17(2):155–179.
- Stabler, E. P. (1997). Derivational minimalism. In Retoré, C., editor, *Logical Aspects of Computational Linguistics (LACL'96)*, volume 1328 of *Lecture Notes in Computer Science*, pages 68–95, New York. Springer.
- Stabler, E. P. (1998). Acquiring languages with movement. *Syntax*, 1(1):72–97.
- Stabler, E. P. (1999). Remnant movement and complexity. *Constraints and resources in natural language syntax and semantics*, 2:299–326.
- Stabler, E. P. (2001a). Minimalist grammars and recognition. In Rohrer, C., Ross-deutscher, A., and Kamp, H., editors, *Linguistic Form and its Computation*. CSLI Publications, Stanford, California.
- Stabler, E. P. (2001b). Recognizing head movement. In Groote, P. D., Morrill, G., and Retoré, C., editors, *Logical Aspects of Computational Linguistics: 4th International Conference, LACL 2001, Le Croisic, France, June 27-29, 2001, Proceedings.*, volume 4, pages 245–260.
- Stabler, E. P. (2006). Sideways without copying. In *Proceedings of the 11th Conference on Formal Grammar*, pages 133–146.



- Stabler, E. P. (2011). Computational perspectives on minimalism. In Boeckx, C., editor, *Oxford Handbook of Linguistic Minimalism*, pages 617–641. Oxford University Press.
- Stabler, E. P. (2013a). The epicenter of linguistic behavior. *Language Down the Garden Path: The Cognitive and Biological Basis of Linguistic Structures*, pages 316–323.
- Stabler, E. P. (2013b). Two models of minimalist, incremental syntactic analysis. *Topics in Cognitive Science*, 5:611–633.
- Stanojević, M. (2016). Minimalist grammar transition-based parsing. In *International Conference on Logical Aspects of Computational Linguistics*, pages 273–290. Springer.
- Stanojević, M. (2019). On the Computational Complexity of Head Movement and Affix Hopping. In *International Conference on Formal Grammar*. Springer.
- Stanojević, M. and Stabler, E. (2018). A sound and complete left-corner parsing for minimalist grammars. In *Proceedings of the Eighth Workshop on Cognitive Aspects of Computational Language Learning and Processing*, pages 65–74.
- Steedman, M. (1996). *Surface Structure and Interpretation*. Linguistic Inquiry Monograph 30. MIT Press, Cambridge, MA.
- Steedman, M. (2000). *The Syntactic Process*. MIT Press, Cambridge, Massachusetts.
- Stowell, T. A. (1981). *Origins of phrase structure*. PhD thesis, Massachusetts Institute of Technology.
- Stroik, T. S. (1996). Extraposition and expletive-movement: A minimalist account. *Lingua*, 99(4):237–251.
- Szabolcsi, A. (2006). Strong vs. weak islands. *The Blackwell companion to syntax*, pages 479–531.
- Takami, K.-i. (1992). *Preposition stranding: From syntactic to functional analyses*, volume 7. Walter de Gruyter.
- Torr, J. (2017). Autobank: a semi-automatic annotation tool for developing deep minimalist grammar treebanks. In *Proceedings of the EACL 2017 Software Demonstrations, Valencia, Spain, April 3-7 2017*, pages 81–86.

- Torr, J. (2018). Constraining MGBank: Agreement, I-selection and supertagging in minimalist grammars. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 590–600.
- Torr, J. and Stabler, E. P. (2016). Coordination in minimalist grammars: Excorporation and across the board (head) movement. In *Proceedings of the Twelfth International Workshop on Tree Adjoining Grammar and Related Formalisms (TAG+12)*, pages 1–17.
- Torr, J., Stanojević, M., Steedman, M., and Cohen, S. (2019). Wide-coverage neural a\* parsing for minimalist grammars. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 2486–2505.
- Vadas, D. and Curran, J. (2007). Adding noun phrase structure to the penn treebank. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 240–247, Prague. ACL.
- Van Riemsdijk, H. and Williams, E. (1981). Np-structure. *The linguistic review*, 1(2):171–218.
- Vergnaud, J.-R. (1974). *French relative clauses*. PhD thesis, Massachusetts Institute of technology.
- Vergnaud, J.-R. (1982). *Dépendances et niveaux de représentation en syntaxe*. PhD thesis, Université de Paris.
- Vinyals, O., Kaiser, Ł., Koo, T., Petrov, S., Sutskever, I., and Hinton, G. (2015). Grammar as a foreign language. In *Advances in neural information processing systems*, pages 2773–2781.
- Warstadt, A., Singh, A., and Bowman, S. R. (2018). Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*.
- Weischedel, R., Pradhan, S., Ramshaw, L., Kaufman, J., Franchini, M., El-Bachouti, M., Xue, N., Palmer, M., Hwang, J. D., Bonial, C., Choi, J., Mansouri, A., Foster, M., aati Hawwary, A., Marcus, M., Taylor, A., Greenberg, C., Hovy, E., Belvin, R., and Houston, A. (2012). *OntoNotes Release 5.0 with OntoNotes DB Tool v0.999 beta*.

- Wexler, K. and Culicover, P. (1980). *Formal Principles of Language Acquisition*. MIT Press, Cambridge, MA.
- Wilder, C. (1997). Some properties of ellipsis in coordination. *Studies on universal grammar and typological variation*, 59:107.
- Woods, W. A. (1970). Transition network grammars for natural language analysis. *Communications of the ACM*, 13(10):591–606.
- Woods, W. A. (1973). An experimental parsing system for transition network grammars. *Natural language processing*, pages 111–154.
- Wu, H., Zhang, J., and Zong, C. (2017). A dynamic window neural network for CCG supertagging. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, pages 3337–3343.
- Xia, F., Palmer, M., and Joshi, A. (2000). A uniform method of grammar extraction and its applications. In *Proceedings of the joint Conference on Empirical Methods in Natural Language Processing and 38th Annual Meeting of the Association for Computational Linguistics*, pages 53–62.
- Xu, W. (2016). LSTM shift-reduce CCG parsing. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1754–1764.
- Xue, N., Xia, F., Chiou, F.-D., and Palmer, M. (2005). The penn chinese treebank: Phrase structure annotation of a large corpus. *Natural Language Engineering*, 11:207–238.
- Yu, K., Miyao, Y., Wang, X., Matsuzaki, T., and Tsujii, J. (2010). Semi-automatically developing chinese hpsg grammar from the penn chinese treebank for deep parsing. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters, COLING '10*, pages 1417–1425, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Zhang, N. N. (2008). Repetitive and correlative coordinators as focus particles parasitic on coordinators. *SKY Journal of Linguistics*, 21:295–342.
- Zhang, N. N. (2010). *Coordination in Syntax, Cambridge Studies in Linguistics Series 123*. Cambridge University Press, Cambridge, UK.

- Zwart, J.-W. (2002). Issues relating to a derivational theory of binding. *Derivation and explanation in the minimalist program*, pages 269–304.
- Zwicky, A. M., Friedman, J., Hall, B. C., and Walker, D. E. (1965). The mitre syntactic analysis procedure for transformational grammars. In *Proceedings of the November 30–December 1, 1965, fall joint computer conference, part I*, pages 317–326. ACM.