5-13-2011

Expressiveness and Succinctness of First-Order Logic on Finite Words

Philipp P. Weis
University of Massachusetts Amherst, pweis@cs.umass.edu

EXPRESSIVENESS AND SUCCINCTNESS OF FIRST-ORDER LOGIC ON FINITE WORDS

A Dissertation Presented

by

PHILIPP WEIS

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2011

Computer Science

EXPRESSIVENESS AND SUCCINCTNESS OF FIRST-ORDER LOGIC ON FINITE WORDS

A Dissertation Presented

by

PHILIPP WEIS

Approved as to style and content by:

Neil Immerman, Chair

David A. Mix Barrington, Member

Rajesh Bhatt, Member

Gerome Miklau, Member

Howard Straubing, Member

Andrew G. Barto, Department Chair
Computer Science

To all my teachers.

ACKNOWLEDGMENTS

Numerous people have been instrumental to the development of this dissertation. Above all, I am grateful for the sustained support and encouragement from my advisor Neil Immerman. He allowed me to develop and explore my own interests, provided consistently helpful and insightful advice, and always had an open ear for my latest ideas or questions. I am particularly fond of him for encouraging me to pick hard problems with unclear outcome and almost guaranteed frustration over more approachable ones of lesser importance.

I want to thank the members of my committee, David A. Mix Barrington, Rajesh Bhatt, Neil Immerman, Gerome Miklau, and Howard Straubing. They asked all the right questions, provided helpful advice, and pushed me further than I would have thought possible.

Additional gratitude is due to Thomas Wilke for bringing to my attention the implications of the structure result for FO^2 on the complexity of the satisfiability for this logic, Michael Benedikt for asking a crucial question on the difference between finite words and finite power words that led to the development of the second chapter of this dissertation, and Qiqi Yan for bringing to my attention a result by Andreas Potthoff on the power of transitive closure logic.

Last but not least, this work would not have been possible without the continued financial support from the NSF through regular grants CCF 0514621 and CCF 0830174.

ABSTRACT

EXPRESSIVENESS AND SUCCINCTNESS OF FIRST-ORDER LOGIC ON FINITE WORDS

MAY 2011

PHILIPP WEIS

M.Sc., UNIVERSITY OF MASSACHUSETTS AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Neil Immerman

Expressiveness, and more recently, succinctness, are two central concerns of finite model theory and descriptive complexity theory. Succinctness is particularly interesting because it is closely related to the complexity-theoretic trade-off between parallel time and the amount of hardware. We develop new bounds on the expressiveness and succinctness of first-order logic with two variables on finite words, present a related result about the complexity of the satisfiability problem for this logic, and explore a new approach to the generalized star-height problem from the perspective of logical expressiveness.

We give a complete characterization of the expressive power of first-order logic with two variables on finite words. Our main tool for this investigation is the classical Ehrenfeucht-Fraïssé game. Using our new characterization, we prove that the quantifier alternation hierarchy for this logic is strict, settling the main remaining open question about the expressiveness of this logic.

A second important question about first-order logic with two variables on finite words is about the complexity of the satisfiability problem for this logic. Previously it was only known that this problem is NP-hard and in NEXP. We prove a polynomial-size small-model property for this logic, leading to an NP algorithm and thus proving that the satisfiability problem for this logic is NP-complete.

Finally, we investigate one of the most baffling open problems in formal language theory: the generalized star-height problem. As of today, we do not even know whether there exists a regular language that has generalized star-height larger than 1. This problem can be phrased as an expressiveness question for first-order logic with a restricted transitive closure operator, and thus allows us to use established tools from finite model theory to attack the generalized star-height problem. Besides our contribution to formalize this problem in a purely logical form, we have developed several example languages as candidates for languages of generalized star-height at least 2. While some of them still stand as promising candidates, for others we present new results that prove that they only have generalized star-height 1.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure **Page**

INTRODUCTION

One of the central concerns of finite model theory is the characterization of the expressiveness of restrictions and extensions of first-order logic on finite structures. Expressiveness is a crucial characteristic of any logic, but expressibility of a given property alone is usually not all that is required – typically we also want to know that the properties of interest can be expressed succinctly. The concept of succinctness has attracted increased attention over the last couple of years, and our interest in it particularly stems from its close connections to complexity theory. For example, the trade-off between the number of variables and formula size in first-order logic corresponds exactly to the trade-off between the number of processors and parallel time.

We develop new bounds on the expressiveness and succinctness of first-order logic with two variables on finite words, present a related result about the complexity of the satisfiability problem for this logic, and explore a new approach to the generalized star-height problem from the perspective of logical expressiveness.

In Chapter 1, we give a complete characterization of the expressive power of first-order logic with two variables on finite words. Our main tool for this investigation is the classical Ehrenfeucht-Fraïssé game. Using this new characterization, we prove that the quantifier alternation hierarchy for this logic is strict, settling the main remaining open question about the expressiveness of this logic [9, 10].

A second important open question about first-order logic with two variables on finite words, which we settle in Chapter 2, is about the complexity of the satisfiability problem for this logic. Previously it was only known that this problem is NP-hard and in NEXP [9, 10]. We prove a polynomial-size small-model property for this logic, leading to an NP algorithm and thus proving that the satisfiability problem for this logic is NP-complete.

Chapter 3 surveys known results and techniques for succinctness of first-order logics. We adapt an established lower bound technique to give a more precise size lower bound for a simple first-order property, and we hope that this approach will prove to be applicable to other first-order properties as well.

Finally, in Chapter 4 we investigate one of the most baffling open problems in formal language theory: the generalized star-height problem [30, 43, 46]. As of today, we do not even know whether there exists a regular language that has generalized star-height larger than 1. This problem can be phrased as an expressiveness question for first-order logic with a restricted transitive closure operator, and thus allows us to use established tools from finite model theory to attack the generalized star-height problem. Besides our contribution to formalize this problem in a purely logical form, we have developed several example languages as candidates for languages of generalized star-height at least 2. While some of them still stand as promising candidates, for others we present new results that prove they only have generalized star-height 1.

I.1 Definitions and Notation

First-order logic is defined in the usual way with boolean connectives $\neg$ and $\vee$, variables $x, y, \ldots$, relation symbols $P, Q, \ldots$, and existential quantification $\exists x$, but not including any function symbols. $\varphi \wedge \psi$ is an abbreviation for $\neg(\neg\varphi \vee \neg\psi)$, $\forall x\, \varphi$ is an abbreviation for $\neg\exists x\, \neg\varphi$, $\top$ is an abbreviation for $\exists x\, x = x$ (always true), and $\bot$ is an abbreviation for $\neg\top$ (always false). $\mathrm{Free}(\varphi)$ is the set of variables that occur freely in φ, i.e. they are not bound by any quantifier. The *size* of a first-order formula φ, denoted by $|\varphi|$, is the number of nodes in its parse tree.

We also use $\exists x.\varphi\,\psi$ as an abbreviation for $\exists x\,(\varphi \wedge \psi)$, and $\forall x.\varphi\,\psi$ as an abbreviation for $\forall x\,(\varphi \rightarrow \psi)$. This notation, together with the usual superscript notation for repetitions, allows us to efficiently represent certain formulas. For example, the formula

$$\exists y.\mathrm{Suc}(x,y)\ [\exists x.(\mathrm{Suc}(y,x)\,\exists y.(\mathrm{Suc}(x,y)]^5\,P(x) \quad ,$$

where Suc is the successor relation and the part in square brackets is literally repeated five times, says that P holds 11 positions to right of x.

We often indicate the free variables of a formula in parenthesis, as in $\varphi(x)$ or $\psi(x,y)$. If $\varphi(x)$ is a formula with x as the only free variable, then $\varphi(y)$ is the same formula with all free occurrences of x replaced by y, and any bound variables renamed as necessary to retain the original meaning of φ. For example, if $\varphi(x) = \exists y\, y > x$, then $\varphi(y) = \exists z\, z > y$.

The *linear order* of size ℓ is the logical structure LO_ℓ with universe $\{1,\ldots,\ell\}$ and a linear order $<$. Depending upon context, LO_ℓ may also interpret a binary successor relation Suc.

Σ will always denote a *finite alphabet*, and Σ_k with $k \in \mathbb{N}^+$ is a finite alphabet with k distinct letters. The empty string is denoted by ε. For a *finite word* $w \in \Sigma^\ell$ and $i \in [1,\ell]$, w_i is the letter at position i of w, and for $[i,j]$ a subinterval of $[1,\ell]$, $w_{[i,j]}$ is the substring $w_i \ldots w_j$. Slightly abusing notation, we identify a finite word $w \in \Sigma^\ell$, $\ell > 0$ with the logical structure $w = (\{1,\ldots,\ell\}, <^w, Q_a^w : a \in \Sigma, x^w, y^w, \ldots)$. Here Q_a are all unary relation symbols, and $x, y, \ldots$ are variables. For all $a \in \Sigma$, we have $Q_a^w = \{1 \leq i \leq \ell \mid w_i = a\}$. Again, depending upon context, w may also interpret a binary successor relation Suc. Thus every finite word is an extension of the underlying linear order.

A structure is *monadic* if all its relations are monadic, not counting numeric relations $<$ and Suc.

For a structure w, we write $|w|$ for the universe of w, and $\|w\|$ for the size of the universe of w.

The default interpretation for all variables is 1, the initial position of the word. For a structure w, a variable x and $i, j \in \|w\|$, $(w, i/x)$ is the structure w with x interpreted as i. We use (w, i) as an abbreviation for $(w, i/x)$, and (w, i, j) as an abbreviation for $(w, i/x, j/y)$.

Whenever convenient, we use constants min and max pointing to the first and last position of a linear order or finite word. Any formula using these constants can easily be converted into a formula that does not use them, at the cost of increasing the quantifier depth by at most 2: We replace $Q_a(\min)$ with $\exists x\, (\forall y\, y \geq x \wedge Q_a(x))$, we replace $x > \min$ by $\exists y\, y < x$, and we replace $x = \min$ by $\neg\exists y\, y < x$. Similar replacements are performed for the constant max.

We use FO[] to denote first-order logic without any numeric predicates, FO[<] to denote first-order logic with a binary linear order predicate $<$, and $\mathrm{FO}[<, \mathrm{Suc}]$ for first-order logic with a binary linear order predicate and an additional binary successor predicate. FO^k is the restriction of first-order logic that uses at most k distinct variables, FO_n refers to the restriction to quantifier depth n, and $\mathrm{FO}_{m,n}$ is the further restriction to formulas such that any path in their parse tree has at most m blocks of alternating quantifiers. Additionally, we set $\mathrm{FO{-}ALT}[m] = \bigcup_{n \geq m} \mathrm{FO}_{m,n}$.

For a class of finite structures $\mathcal{C}$ and a logical formula φ, we write $\mathsf{Mod}_{\mathcal{C}}(\varphi)$ for the set of all $\mathcal{C}$-structures that satisfy φ. As usual, when $\mathcal{C}$ is clear from the context, we identify L with the set of properties expressible in L on $\mathcal{C}$, i.e. $L = \{\mathsf{Mod}_{\mathcal{C}}(\varphi) \mid \varphi \in L\}$. For two structures w and w' and a logic L, w and w' are *L-equivalent*, $w \equiv'_L w'$, if w and w' agree on all L-formulas. For more convenient notation, we write $w \equiv^k_n w'$ when w and w' agree on all formulas from FO^k_n, and $w \equiv^k_{m,n} w'$ if they agree on $\mathrm{FO}^k_{m,n}$. Two formulas $\varphi, \psi \in L$ are *$\mathcal{C}$-equivalent*, $\varphi \equiv_{\mathcal{C}} \psi$, if they agree on all $\mathcal{C}$-structures, i.e. $\mathsf{Mod}_{\mathcal{C}}(\varphi) = \mathsf{Mod}_{\mathcal{C}}(\psi)$. When $\mathcal{C}$ is understood from the context, we simply write $\varphi \equiv \psi$.

The language $\mathrm{FO}^2[<, \mathrm{Suc}]$ is more expressive than $\mathrm{FO}^2[<]$ because it allows us to talk about consecutive strings of symbols. With three variables we can express $\mathrm{Suc}(x, y)$ using the ordering: $x < y \wedge \forall z\,(z \leq x \vee y \leq z)$. Thus for any $k > 2$, $\mathrm{FO}^k[<, \mathrm{Suc}] = \mathrm{FO}^k[<]$.

I.2 Ehrenfeucht-Fraïssé Games

Ehrenfeucht-Fraïssé games are one of our main tools. They are two-player games played on two logical structures, where one player (Samson, the spoiler) tries to point out a difference in the two structures by placing a pebble on an element of one of the two structures, and the other player (Delilah, the duplicator) replies by placing a corresponding pebble on an element of the other structure with the goal of making the two structures look the same.

For some structures w and w', and $k \in \mathbb{N}^+$ and $n \in \mathbb{N}$, $\mathrm{FO}^k_n(w, w')$ is the k-pebble n-move game on w and w'. Each one of the players has k colored or named pebbles. In every move of the game, Samson places one of his pebbles on a position in one of the two words, and Delilah answers by placing her pebble of the same color on a position of the other word. Samson wins if after some move, the map from the chosen points in w to those in w' is not an isomorphism of the induced substructures; and Delilah wins otherwise. The fundamental theorem of Ehrenfeucht-Fraïssé games is the following:

Theorem I.2.1. [21] Let $w, w' \in \Sigma^\star$, $k \in \mathbb{N}^+$ and $n \in \mathbb{N}$. Delilah has a winning strategy for the game $\mathrm{FO}^k_n(w, w')$ iff $w \equiv^k_n w'$.

Ehrenfeucht-Fraïssé games are determined, meaning that in any game exactly one of the two players has a winning strategy. Since there are only finitely many inequivalent formulas in FO^k_m [21], Theorem I.2.1 provides a complete characterization of first-order expressiveness. Furthermore, in the case where Samson has a winning strategy, the game tree for his strategy corresponds exactly to the distinguishing formula. For more details on Ehrenfeucht-Fraïssé games, we refer the reader to [21].

The game $\mathrm{FO}^k_{m,n}(w, w')$ is the restriction of the game $\mathrm{FO}^k_n(w, w')$ in which Samson may change which word he plays on at most $m - 1$ times.

Theorem I.2.2. [21] Let $w, w' \in \Sigma^\star$ and let $k \in \mathbb{N}+$, $m, n \in \mathbb{N}$ with $m \leq n$. Delilah has a winning strategy for the game $\mathrm{FO}^2_{m,n}(w, w')$ iff $w \equiv^2_{m,n} w'$.

We end this section with a simple lemma that will be useful whenever we want to prove that there is a formula expressing a property of strings. With this lemma, it suffices to show that for any pair of strings, one with the property in question and one without, there is a formula that distinguishes between these two particular strings.

Lemma I.2.3. Let $P \subseteq \Sigma^\star$ and let L be a logic closed under boolean operations with only finitely many inequivalent formulas. If for every $w \in P$ and every $w' \in \overline{P}$ there is a formula $\varphi_{w,w'} \in L$ such that $w \models \varphi_{w,w'}$ and $w' \not\models \varphi_{w,w'}$, then there is a formula $\varphi \in L$ such that for all $w \in \Sigma^\star$, $w \models \varphi$ iff $w \in P$.

Proof. Let $\Gamma := \{\varphi_{w,w'} \mid w \in P, w' \in \overline{P}\}$, and let Γ' be a maximal subset of Γ containing only inequivalent formulas. Since L contains only finitely many inequivalent formulas, Γ' is finite. For every $w \in P$, we define the finite sets of formulas $\Gamma'_w := \{\varphi \in \Gamma' \mid w \models \varphi\}$. Since all these sets are subsets of the finite set Γ', there can only be finitely many of them. Thus there is a finite set $P' \subseteq P$ such that $\{\Gamma'_w \mid w \in P\} = \{\Gamma'_w \mid w \in P'\}$. Now we set

$$\varphi := \bigvee_{w \in P'} \bigwedge_{\varphi \in \Gamma'_w} \varphi \quad .$$

We have $\varphi \in L$ and for every $w \in \Sigma^\star$, $w \in P$ iff $w \models \varphi$ as required. $\square$

It is well-known that for any $k \in \mathbb{N}^+$ and $m, n \in \mathbb{N}$, the logics FO^k_n and $\mathrm{FO}^k_{m,n}$, both with and without the successor predicate, have only finitely many inequivalent formulas [21]. Thus the above lemma applies to these logics.

CHAPTER 1

STRUCTURE THEOREM FOR FO^2 ON FINITE WORDS

It is well-known that every first-order property on words is expressible using at most three variables [23, 24]. The subclass of properties expressible with only two variables is also quite interesting and well-studied (Theorem 1.0.4).

We prove precise structure theorems that characterize the exact expressive power of first-order logic with two variables on finite words. Our results apply to $\mathrm{FO}^2[<]$ and $\mathrm{FO}^2[<, \mathrm{Suc}]$. For both languages, our structure theorems show exactly what is expressible using a given quantifier depth, n, and using m blocks of alternating quantifiers, for any $m \leq n$. Using these characterizations, we prove that there is a strict hierarchy of alternating quantifiers for both languages. The question whether there was such a hierarchy had been completely open since it was asked in [9, 10].

Our characterization of $\mathrm{FO}^2[<]$ and $\mathrm{FO}^2[<, \mathrm{Suc}]$ on finite words is based on the very natural notion of n-ranker (Definition 1.1.2). Informally, a ranker is the position of a certain combination of letters in a finite word. For example, $\triangleright_{\mathtt{a}}$ and $\triangleleft_{\mathtt{b}}$ are 1-rankers where $\triangleright_{\mathtt{a}}(w)$ is the position of the first $\mathtt{a}$ in w (from the left) and $\triangleleft_{\mathtt{b}}(w)$ is the position of the first $\mathtt{b}$ in w from the right. Similarly, the 2-ranker $r_2 = \triangleright_{\mathtt{a}}\triangleright_{\mathtt{c}}$ denotes the position of the first $\mathtt{c}$ to the right of the first $\mathtt{a}$, and the 3-ranker, $r_3 = \triangleright_{\mathtt{a}} \triangleright_{\mathtt{c}} \triangleleft_{\mathtt{b}}$ denotes the position of the first $\mathtt{b}$ to the left of r_2. If there is no such letter then the ranker is undefined. For example, $r_3(\mathtt{cababcba}) = 5$ and $r_3(\mathtt{acbbca})$ is undefined.

Our first structure theorem (Theorem 1.1.8) says that the properties expressible in $\mathrm{FO}^2_n[<]$, i.e. first-order logic with two variables and quantifier depth n, are exactly boolean combinations of statements of the form, "r is defined", and "r is to the left (right) of r'" for k-rankers, r, and k'-rankers, r', with $k \leq n$ and $k' < n$. A non-quantitative version of this theorem in terms of "turtle languages" was previously known [34]. Furthermore, a quantitative version in terms of iterated block products of the variety of semi-lattices is presented in [40], based on work by Straubing and Thérien [37].

For $\mathrm{FO}^2[<, \mathrm{Suc}]$, a straightforward generalization of n-ranker to n-successor-ranker allows us to prove exact analogs of Theorems 1.1.8 and 1.2.5. We use the latter to prove that there is also a strict alternation hierarchy for $\mathrm{FO}^2_n[<, \mathrm{Suc}]$ (Theorem 1.3.6). Since in the presence of successor we can encode an arbitrary alphabet in binary, no analog of Theorem 1.2.7 holds for $\mathrm{FO}^2[<, \mathrm{Suc}]$.

Surprisingly, Theorem 1.1.8 can be generalized in almost exactly the same form to characterize $\mathrm{FO}^2_{m,n}[<]$ where there are at most m blocks of alternating quantifiers, $m \leq n$. This second structure theorem (Theorem 1.2.5) uses the notion of (m, n)-ranker where there are m blocks of $\triangleright$'s or $\triangleleft$'s, that is, changing direction in rankers corresponds exactly to alternation of quantifiers. Using Theorem 1.2.5 we prove that there is a strict alternation hierarchy for $\mathrm{FO}^2_n[<]$ (Theorem 1.2.11) but that exactly at most $|\Sigma| + 1$ alternations are useful, where $|\Sigma|$ is the size of the alphabet (Theorem 1.2.7).

Many beautiful results on FO^2 on finite words were already known. The main significant outstanding question was whether there was an alternation hierarchy. The following is a summary of the main previously known characterizations of $\mathrm{FO}^2[<]$ on finite words. For a detailed treatment of all these characterizations, we refer the reader to [39].

Theorem 1.0.4. $[9, 10, 31, 33, 34, 42]$ Let $R \subseteq \Sigma^\star$. The following statements are equivalent:

(1) $R \in \mathrm{FO}^2[<]$.

(2) R is expressible in unary temporal logic.

(3) $R \in \Sigma_2 \cap \Pi_2[<]$.

(4) R is an unambiguous regular language.

(5) The syntactic semi-group of R is a member of **DA**.

(6) R is recognizable by a partially-ordered 2-way automaton.

(7) R is a boolean combination of “turtle languages”.

The proofs of our structure theorems are self-contained applications of Ehrenfeucht-Fraïssé games. All of the above characterizations follow from these results. Furthermore, we have now exactly connected quantifier and alternation depth to the picture, thus adding tight bounds and further insight to the above results.

For example, one can best understand item 4 above – that $\mathrm{FO}^2[<]$ on finite words corresponds to the unambiguous regular languages – via Theorem 1.1.12 which states that any $\mathrm{FO}^2_n[<]$ formula with one free variable that is always true of at most one position in any string, necessarily denotes an n-ranker.

In the conclusion of [34], the authors define the subclasses of rankers with one and two blocks of alternation. They write that,

> “[...] turtle languages might turn out to be a helpful tool for further studies in algebraic language theory.”

We feel that the results in this chapter fully justify that prediction. Turtle languages — aka rankers — do provide an exceptionally clear and precise understanding of the expressive power of FO^2 on finite words, with and without successor.

In summary, our structure theorems provide a complete classification of the expressive power of FO^2 on finite words in terms of both quantifier depth and alternation. They also tighten several previous characterizations and lead to the alternation hierarchy results.

In Section 1.1 we formally define rankers and present our structure theorem for $\mathrm{FO}^2_n[<]$. The structure theorem for $\mathrm{FO}^2_{m,n}[<]$ is covered in Section 1.2, including our alternation hierarchy result that follows from it. Section 1.3 extends our structure theorems and the alternation hierarchy result to $\mathrm{FO}^2[<, \mathrm{Suc}]$.

1.1 Structure Theorem for $\mathrm{FO}^2[<]$

We define boundary positions that point to the first or last occurrences of a letter in a word, and define an n-ranker as a sequence of n boundary positions. In terms of [34], boundary positions are turtle instructions and n-rankers are turtle programs of length n. The following three lemmas show that basic properties about the definedness and position of these rankers can be expressed in $\mathrm{FO}^2[<]$, and we use these results to prove our structure theorem.

Definition 1.1.1. A *boundary position* denotes the first or last occurrence of a letter in a given word. Boundary positions are of the form $d_{\mathtt{a}}$ where $d \in \{\triangleright, \triangleleft\}$ and $\mathtt{a} \in \Sigma$. The interpretation of a boundary position $d_{\mathtt{a}}$ on a word $w = w_1 \ldots w_{\|w\|} \in \Sigma^\star$ is defined as follows.

$$d_{\mathtt{a}}(w) := \begin{cases} \min\{i \in |w| \mid w_i = \mathtt{a}\} & \text{if } d = \triangleright \\ \max\{i \in |w| \mid w_i = \mathtt{a}\} & \text{if } d = \triangleleft \end{cases}$$

Here we set $\min\{\}$ and $\max\{\}$ to be undefined, thus $d_{\mathtt{a}}(w)$ is undefined if a does not occur in w. A boundary position can also be specified with respect to a position $q \in |w|$.

$$d_{\mathtt{a}}(w, q) := \begin{cases} \min\{i \in [q+1, \|w\|] \mid w_i = \mathtt{a}\} & \text{if } d = \triangleright \\ \max\{i \in [1, q-1] \mid w_i = \mathtt{a}\} & \text{if } d = \triangleleft \end{cases}$$

Definition 1.1.2. Let n be a positive integer. An *n-ranker* r is a sequence of n boundary positions. The interpretation of an n-ranker $r = (p_1, \ldots, p_n)$ on a word w is defined as follows.

$$r(w) := \begin{cases} p_1(w) & \text{if } r = (p_1) \\ \text{undefined} & \text{if } (p_1, \ldots, p_{n-1})(w) \text{ is undefined} \\ p_n(w, (p_1, \ldots, p_{n-1})(w)) & \text{otherwise} \end{cases}$$

Instead of writing n-rankers as a formal sequence $(p_1, \ldots, p_n)$, we often use the simpler notation $p_1 \ldots p_n$. We denote the set of all n-rankers by R_n, and the set of all n-rankers that are defined over a word w by $R_n(w)$. Furthermore, we set $R_n^\star := \bigcup_{i \in [1,n]} R_i$ and $R_n^\star(w) := \bigcup_{i \in [1,n]} R_i(w)$.

Definition 1.1.3. Let r be an n-ranker. As defined above, we have $r = (p_1, \ldots, p_n)$ for boundary positions p_i. The *k-prefix ranker* of r for $k \in [1, n]$ is $r_k := (p_1, \ldots, p_k)$.

Definition 1.1.4. Let $i, j \in \mathbb{N}$. The *order type* of i and j is defined as

$$\operatorname{ord}(i, j) := \begin{cases} < & \text{if } i < j \\ = & \text{if } i = j \\ > & \text{if } i > j \end{cases} .$$

Lemma 1.1.5 (distinguishing points on opposite sides of a ranker)**.** Let n be a positive integer, let $w, w' \in \Sigma^\star$ and let $r \in R_n(w) \cap R_n(w')$. Samson wins the game $\mathrm{FO}_n^2(w, w')$ where initially $\operatorname{ord}(x^w, r(w)) \neq \operatorname{ord}(x^{w'}, r(w'))$.

Proof. We only look at the case where $x^w \geq r(w)$ and $x^{w'} < r(w')$ since all other cases are symmetric to this one. For $n = 1$ Samson has a winning strategy: If r is the first occurrence of a letter, then Samson places y on $r(w)$ and Delilah cannot reply. If r marks the last occurrence of a letter in the whole word, then Samson places y on $r(w')$. Again, Delilah cannot reply with any position and thus loses.

For $n > 1$, we look at the prefix ranker r_{n-1} of r. If $r_{n-1}(w) < r(w)$, as shown in Figure 1.1, then Samson places pebble y on $r(w)$. Delilah has to reply with a position that is to the left of $x^{w'}$. She cannot choose a position in the interval $(r_{n-1}(w'), r(w'))$, because this section does not contain the letter $w_{r(w)}$. Thus she has to choose a position left of or equal to $r_{n-1}(w')$. By induction Samson wins the remaining game.

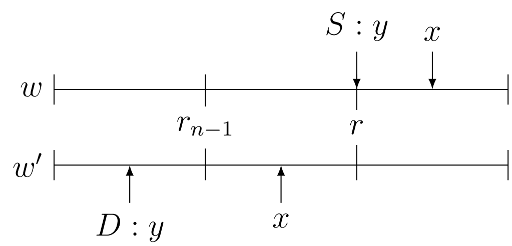

Figure 1.1. Proof of Lemma 1.1.5: $r_{n-1}(w) < r(w)$.

Otherwise we have $r(w) < r_{n-1}(w)$, as shown in Figure 1.2. Samson places y on $r(w')$, and Delilah has to reply with a position to the right of x^w and thus to the right of $r(w)$. She cannot choose any position in $(r(w), r_{n-1}(w))$, because this interval does not contain the letter $w'_{r(w')}$, thus Delilah has to choose a position to the right of or equal to $r_{n-1}(w)$. By induction Samson wins the remaining game. $\square$

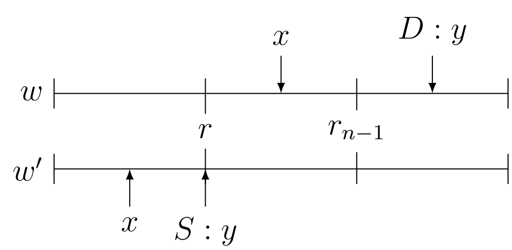

Figure 1.2. Proof of Lemma 1.1.5: $r(w) < r_{n-1}(w)$.

Lemma 1.1.6 (expressing the definedness of a ranker)**.** Let n be a positive integer, and let $r \in R_n$. There is a formula $\varphi_r \in \mathrm{FO}^2_n[<]$ such that for all $w \in \Sigma^\star$, $w \models \varphi_r$ iff $r \in R_n(w)$.

Proof. Using Lemma $I.2.3$ it suffices to consider arbitrary $w, w' \in \Sigma^\star$ with $r \in R_n(w)$ and $r \notin R_n(w')$, and using Theorem I.2.1, it suffices to show that Samson wins the game $\mathrm{FO}^2_n(w, w')$. If r_1, the shortest prefix ranker of r, is not defined over w', the letter referred to by r_1 occurs in w but does not occur in w'. Thus Samson easily wins in one move.

Otherwise we let $r_i = (p_1, \ldots, p_i)$ be the shortest prefix ranker of r that is undefined over w'. Thus r_{i-1} is defined over both words. Without loss of generality we assume that $p_i = \triangleleft_{\mathtt{a}}$. This situation is illustrated in Figure 1.3. Notice that w' does not contain any $\mathtt{a}$'s to the left of $r_{i-1}(w')$, otherwise r_i would be defined over w'. Samson places x in w on $r_i(w)$, and Delilah has to reply with a position right of or equal to $r_{i-1}(w')$. Now Lemma 1.1.5 applies and Samson wins in $i-1$ more moves. $\square$

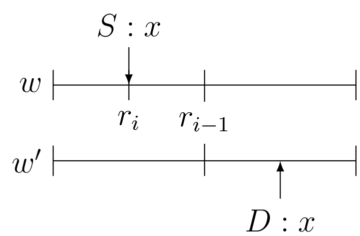

Figure 1.3. Proof of Lemma 1.1.6: $r_i(w')$ is undefined.

Lemma 1.1.7 (position of a ranker)**.** Let n be a positive integer and let $r \in R_n$. There is a formula $\psi_r \in \mathrm{FO}^2_n[<]$ such that for all $w \in \Sigma^\star$ and for all $i \in |w|$, $(w, i) \models \psi_r$ iff $i = r(w)$.

Proof. As in the proof of Lemma 1.1.6, it suffices to show that for arbitrary $w, w' \in \Sigma^\star$, Samson wins the game $\mathrm{FO}^2_n(w, w')$ where initially $x^w = r(w)$ and $x^{w'} \neq r(w')$. If $r(w')$ is defined over w', then we can apply Lemma 1.1.5 immediately to get the desired strategy for Samson. Otherwise we use the strategy from Lemma 1.1.6. $\square$

Theorem 1.1.8 (structure of $\mathrm{FO}^2_n[<]$)**.** Let w and w' be finite words, and let $n \in \mathbb{N}$. The following two conditions are equivalent.

(i) (a) $R_n(w) = R_n(w')$, and,

(b) for all $r \in R^\star_n(w)$ and $r' \in R^\star_{n-1}(w)$, $\mathrm{ord}(r(w), r'(w)) = \mathrm{ord}(r(w'), r'(w'))$

(ii) $w \equiv^2_n w'$

Notice that condition (i)(a) is equivalent to $R^\star_n(w) = R^\star_n(w')$. Instead of proving Theorem 1.1.8 directly, we prove the following more general version on words with two interpreted variables.

Theorem 1.1.9. Let w and w' be finite words, let $i_1, i_2 \in |w|$, let $j_1, j_2 \in |w'|$, and let $n \in \mathbb{N}$. The following two conditions are equivalent.

(i) (a) $R_n(w) = R_n(w')$, and,

(b) for all $r \in R^\star_n(w)$ and $r' \in R^\star_{n-1}(w)$, $\mathrm{ord}(r(w), r'(w)) = \mathrm{ord}(r(w'), r'(w'))$, and,

(c) $(w, i_1, i_2) \equiv^2_0 (w', j_1, j_2)$, and,

(d) for all $r \in R^\star_n(w)$, we have $\mathrm{ord}(i_1, r(w)) = \mathrm{ord}(j_1, r(w'))$ and $\mathrm{ord}(i_2, r(w)) = \mathrm{ord}(j_2, r(w'))$

(ii) $(w, i_1, i_2) \equiv^2_n (w', j_1, j_2)$

Proof. For $n = 0$, (i)(a), (i)(b) and (i)(d) are vacuous, and (i)(c) is equivalent to (ii). For $n \geq 1$, we prove the two implications individually using induction on n.

We first show "$\neg$(i) $\Rightarrow$ $\neg$(ii)". Assuming that (i) holds for $n \in \mathbb{N}$ but fails for $n+1$, we show that $(w, i_1, i_2) \not\equiv^2_{n+1} (w', j_1, j_2)$ by giving a winning strategy for Samson in the FO^2_{n+1} game on the two structures. If (i)(c) does not hold, then Samson wins immediately. If (i)(d) does not hold for $n+1$, then Samson wins by Lemma 1.1.5. If (i)(a) or (i)(b) do not hold for $n+1$, then one of the following three cases applies.

(1) There is an $(n+1)$-ranker that is defined over one word but not over the other.

(2) There are two n-rankers that do not agree on their ordering in w and w'.

(3) There is an $(n+1)$-ranker that does not appear in the same order on both structures with respect to a k-ranker where $k \le n$.

We first look at case (2) where there are two rankers $r, r' \in R_n^\star(w)$ that disagree on their ordering in w and w'. Without loss of generality we assume that $r(w) \le r'(w)$ and $r(w') > r'(w')$, and present a winning strategy for Samson in the FO^2_{n+1} game. In the first move he places x on $r(w)$ in w. Delilah has to reply with $r(w')$ in w', otherwise she would lose the remaining n-move game as shown in Lemma 1.1.5. Let r'_{n-1} be the $(n-1)$-prefix-ranker of r'. We look at two different cases depending on the ordering of r'_{n-1} and r'.

For $r'_{n-1}(w) < r'(w)$, the situation is illustrated in Figure 1.4. In his second move, Samson places y on $r'(w')$. Delilah has to reply with a position to the left of x^w, but she cannot choose any position from the interval $(r'_{n-1}(w), r'(w))$ because it does not contain the letter $w'_{y^{w'}}$. So she has to reply with a position left of or equal to $r'_{n-1}(w)$, and Samson wins the remaining FO^2_{n-1} game as shown in Lemma 1.1.5.

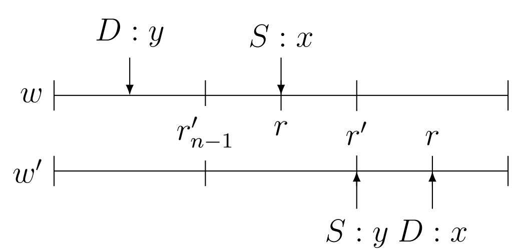

Figure 1.4. Proof of Theorem 1.1.9: Two n-rankers appear in different order and r' ends with $\triangleright$.

For $r'_{n-1}(w) > r'(w)$, the situation is illustrated in Figure 1.5. In his second move, Samson places pebble y on $r'(w)$, and Delilah has to reply with a position to the right of $x^{w'}$, but she cannot choose anything from the interval $(r'(w'), r'_{n-1}(w'))$ because this section does not contain the letter w_{y^w}. Thus she has to reply with a position right of or equal to $r'_{n-1}(w')$, and Samson wins the remaining FO^2_{n-1} game as shown in Lemma 1.1.5.

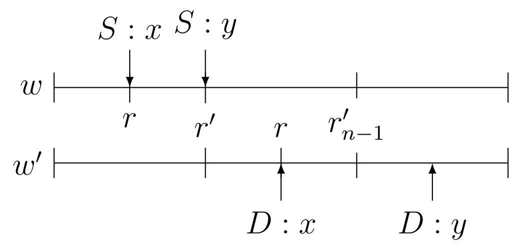

Figure 1.5. Proof of Theorem 1.1.9: Two n-rankers appear in different order and r' ends with $\triangleleft$.

Now we look at cases (1) and (3), assuming that case (2) does not apply. We know that condition (i) from the statement of the theorem fails, but still all n-rankers agree on their ordering. In both case (1) and case (3), there are two consecutive n-rankers $r, r' \in R_n(w)$ with $r(w) < r'(w)$ and a letter $\mathtt{a} \in \Sigma$ such that without loss of generality $\mathtt{a}$ occurs in the segment $w_{((r(w),r'(w))}$ but not in the segment $w'_{(r(w'),r'(w'))}$. We describe a winning strategy for Samson in the game $\mathrm{FO}^2_{n+1}(w, w')$. He places x on an $\mathtt{a}$ in the segment $(r(w), r'(w))$ of w, as shown in Figure 1.6. Delilah cannot reply with anything in the interval $(r(w'), r'(w'))$. If she replies with a position left of or equal to $r(w')$, then x is on different sides of the n-ranker r in the two words. Thus Lemma 1.1.5 applies and Samson wins the remaining n-move game. If Delilah replies with a position right of or equal to $r'(w')$, then we can apply Lemma 1.1.5 to r' and get a winning strategy for the remaining game as well. This concludes the proof of "$\neg$(i) $\Rightarrow$ $\neg$(ii)".

To show "(i) $\Rightarrow$ (ii)", we assume (i) for $n+1$, and present a winning strategy for Delilah in the FO^2_{n+1} game on the two structures. In his first move Samson picks up one of the two pebbles, and places it on a new position. Without loss of generality we assume that Samson picks up x and places it on w in his first move. If $x^w = r(w)$

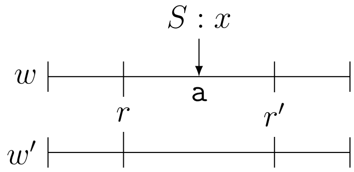

Figure 1.6. Proof of Theorem 1.1.9: A letter $\mathtt{a}$ occurs between n-rankers r, r' in w but not in w'.

for any ranker $r \in R^\star_{n+1}(w)$, then Delilah replies with $x^{w'} = r(w')$. This establishes (i)(c) and (i)(d) for n, and thus Delilah has a winning strategy for the remaining FO^2_n game by induction.

If Samson does not place x^w on any ranker from $R^\star_{n+1}(w)$, then we look at the closest rankers from $R^\star_n(w)$ to the left and right of x^w, denoted by λ and ρ, respectively. Let $\mathtt{a} := w_{x^w}$ and define the $(n+1)$-ranker $s := (\lambda, \rhd_{\mathtt{a}})$. On w we have $\lambda(w) < s(w) < \rho(w)$. Because of (i)(a) s is defined on w' as well, and because of (i)(b), we have $\lambda(w') < s(w') < \rho(w')$. If y^w is not contained in the interval $(\lambda(w), \rho(w))$, then Delilah places x on $s(w')$, which establishes (i)(c) and (i)(d) for n. Thus by induction Delilah has a winning strategy for the remaining FO^2_n game.

If both pebbles x^w and y^w occur in the interval $(\lambda(w), \rho(w))$, then we need to be more careful. Without loss of generality we assume $y^w < x^w$ as illustrated in Figure 1.7. Thus Delilah has to place x in the interval $(y^{w'}, \rho(w'))$ and at a position with letter $\mathtt{a} := w_{x^w}$. We define the $n+1$-ranker $s := (\rho, \lhd_{\mathtt{a}})$. From (i)(d) we know that s appears on the same side of y in both structures, thus we have $y^{w'} < s(w') < \rho(w')$. Delilah places her pebble x on $s(w')$, and thus establishes (i)(c) and (i)(d) for n. By induction, Delilah has a winning strategy for the remaining FO^2_n game. $\square$

A fundamental property of an n-ranker is that it uniquely describes a position in a given word. Now we show that the converse holds as well: Any position in a word that can be uniquely described with an $\mathrm{FO}^2[<]$ formula can also be described by a ranker

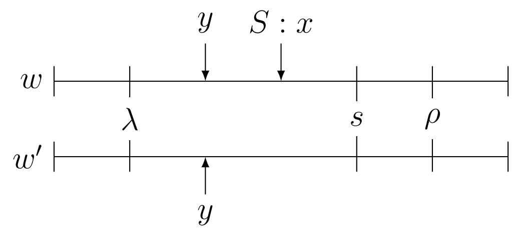

Figure 1.7. Proof of Theorem 1.1.9: x and y are in the same section.

(Lemma 1.1.11). Furthermore, any $\mathrm{FO}^2[<]$ formula that describes a unique position in any given word is equivalent to a boolean combination of rankers (Theorem 1.1.12).

Definition 1.1.10. A formula $\varphi(x) \in \mathrm{FO}^2[<]$ is a *unique position formula* if for all $w \in \Sigma^\star$ there is at most one $i \in |w|$ such that $(w,i) \models \varphi(x)$.

Lemma 1.1.11. Let n be a positive integer and let $\varphi \in \mathrm{FO}^2_n[<]$ be a unique position formula. Let $w \in \Sigma^\star$ and let $i \in |w|$ such that $(w,i) \models \varphi$. Then $i = r(w)$ for some ranker $r \in R_n^\star$.

Proof. Suppose for the sake of a contradiction that there is no ranker $r \in R_n^\star$ such that $(w,i) \models \varphi_r$. Because the first and last positions in w are described by 1-rankers, we know that $i \notin |w|$. We construct a new word w' by doubling the symbol at position i in w, $w' := w_1 \ldots w_{i-1} w_i w_i w_{i+1} \ldots w_{\|w\|}$. By assumption, there is no n-ranker that describes position i in w. A brief argument by contradiction shows that there are also no n-rankers that describe positions i or $i+1$ in w': Assuming that such a ranker exists, let r be the shortest such ranker. Thus none of the prefix rankers of r point to either positions i or $i+1$ in w'. This means that all prefix rankers of r are interpreted in exactly the same way on both w and w', and irrespective of whether $r(w')$ points to i or $i+1$, we have have $r(w) = i$, a contradiction. Hence all n-rankers are insensitive to the doubling of w_i, and the two words w and w' agree on the definedness of all n-rankers and on their ordering. By Theorem 1.1.9, we thus have $(w,i) \equiv_n^2 (w',i) \equiv_n^2 (w',i+1)$, which contradicts the fact that φ is a unique position formula. $\square$

Theorem 1.1.12. Let n be a positive integer and let $\varphi \in \mathrm{FO}_n^2[<]$ be a unique position formula. There is a $k \in \mathbb{N}$, and there are mutually exclusive formulas $\alpha_i \in \mathrm{FO}_n^2[<]$ and rankers $r_i \in R_n^\star$ such that

$$\varphi \equiv \bigvee_{i \in [1,k]} (\alpha_i \wedge \varphi_{r_i}) \quad ,$$

where $\varphi_{r_i} \in \mathrm{FO}_n^2[<]$ is the formula from Lemma 1.1.7 that uniquely describes the ranker r_i.

Proof. Let $\mathcal{T}$ be the set of all $\mathrm{FO}_n^2[<]$ types of words over Σ with one interpreted variable. Because there are only finitely many inequivalent formulas in $\mathrm{FO}_n^2[<]$, $\mathcal{T}$ is finite. Let $\mathcal{T}' \subseteq \mathcal{T}$ be the set of all types that satisfy φ. We set $\mathcal{T}' := \{T_1, \ldots, T_k\}$ and let $\alpha_i \in \mathrm{FO}_n^2[<]$ be a description of type T_i. Thus $\varphi \equiv \bigvee_{i \in [1,k]} \alpha_i$.

Now suppose that $(w,j) \models \varphi$. Thus $(w,j) \models \alpha_i$ for some i. By Lemma 1.1.11 $(w,j) \models \varphi_{r_i}$ for some $r_i \in R_n^\star$. Thus $\alpha_i \to \varphi_{r_i}$ since $\varphi_{r_i} \in \mathrm{FO}_n^2$ and α_i is a complete FO_n^2 formula. Thus $\alpha_i \equiv \alpha_i \wedge \varphi_{r_i}$ so φ is in the desired form. $\square$

1.2 Alternation Hierarchy for $\mathbf{FO}^2[<]$

We define alternation rankers and prove our structure theorem (Theorem 1.2.5) for $\mathrm{FO}_{m,n}^2[<]$. Surprisingly the number of alternating blocks of $\triangleleft$ and $\triangleright$ in the rankers corresponds exactly to the number of alternating quantifier blocks. The main ideas from our proof of Theorem 1.1.8 still apply here, but keeping track of the number of alternations does add complications.

Definition 1.2.1. Let $m, n \in \mathbb{N}$ with $m \le n$. An *m-alternation n-ranker*, or (m, n)-ranker, is an n-ranker with exactly m blocks of boundary positions that alternate between $\triangleright$ and $\triangleleft$.

We use the following notation for alternation rankers.

$$
\begin{aligned}
R_{m,n}(w) &:= \{r \mid r \text{ is an } m\text{-alternation } n\text{-ranker and defined over the word } w\} \\
R_{m\triangleright,n}(w) &:= \{r \in R_{m,n}(w) \mid r \text{ ends with } \triangleright\} \\
R^{\star}_{m,n}(w) &:= \bigcup_{i\in[1,m], j\in[1,n]} R_{i,j}(w) \\
R^{\star}_{m\triangleright,n}(w) &:= R^{\star}_{m-1,n}(w) \cup \bigcup_{i\in[1,n]} R_{m\triangleright,i}(w)
\end{aligned}
$$

Lemma 1.2.2. Let $m, n \in \mathbb{N}^+$ with $m \le n$, let $w, w' \in \Sigma^\star$ be finite words, and let $r \in R_{m,n}(w) \cap R_{m,n}(w')$. Samson wins the game $\mathrm{FO}^2_{m,n}(w, w')$ where initially we have $\mathrm{ord}(r(w), x^w) \ne \mathrm{ord}(r(w'), x^{w'})$.

Furthermore, Samson can start the game with a move on w if r ends with $\triangleright$, $r(w) \le x^w$ and $r(w') \ge x^{w'}$, or if r ends with $\triangleleft$, $r(w) \ge x^w$ and $r(w') \le x^{w'}$. He can start the game with a move on w' if r ends with $\triangleright$, $r(w) \ge x^w$ and $r(w') \le x^{w'}$, or if r ends with $\triangleleft$, $r(w) \le x^w$ and $r(w') \ge x^{w'}$.

Proof. If $m = n = 1$, then we can immediately apply the base case from the proof of Lemma 1.1.5. Samson wins in one move, placing his pebble on w or w' as specified.

For the remaining cases, we assume without loss of generality that r ends with $\triangleright$ and that $x^w \ge r(w)$ and $x^{w'} \le r(w')$. Let r_{n-1} be the $(n-1)$-prefix ranker of r. This situation is illustrated in Figure 1.1 of Lemma 1.1.5. Samson places y on $r(w)$, and creates a situation where $y^w > r_{n-1}(w)$ and $y^{w'} \le r_{n-1}(w')$. If r_{n-1} ends with $\triangleleft$, then by induction Samson wins the remaining $\mathrm{FO}^2_{m-1,n-1}$ game and thus he has a winning strategy for the $\mathrm{FO}^2_{m,n}$ game. If r_{n-1} ends with $\triangleright$, then by induction Samson wins the remaining $\mathrm{FO}^2_{m,n-1}$ game starting with a move on w, and thus he has a winning strategy for the $\mathrm{FO}^2_{m,n}$ game. $\square$

Lemma 1.2.3. Let $m, n \in \mathbb{N}^+$ with $m \leq n$, and let $r \in R_{m,n}$. There is a formula $\varphi_r \in \mathrm{FO}^2_{m,n}[<]$ such that for all $w \in \Sigma^\star$, $w \models \varphi_r$ iff $r \in R_{m,n}(w)$.

Proof. Using Lemma *I*.2.3 it suffices to consider arbitrary $w, w' \in \Sigma^\star$ with $r \in R_{m,n}(w)$ and $r \notin R_{m,n}(w')$, and using Theorem I.2.1, it suffices to show that Samson wins the game $\mathrm{FO}^2_{m,n}(w, w')$. Let $r_i = (p_1, \ldots, p_i)$ be the shortest prefix ranker of r that is undefined over w', and we assume without loss of generality that this ranker ends with the boundary position $p_i = \lhd_{\mathtt{a}}$ for some $\mathtt{a} \in \Sigma$. This situation is illustrated in Figure 1.3 for Lemma 1.1.7. In his first move Samson places x on $r_i(w)$ and thus forces a situation where $x^w < r_{i-1}(w)$ and $x^{w'} \geq r_{i-1}(w')$. If r_{i-1} ends with $\lhd$, then according to Lemma 1.2.2, Samson wins the remaining $\mathrm{FO}^2_{m,n-1}$ game starting with a move on w. Otherwise r_{i-1} ends with $\rhd$, and thus by Lemma 1.2.2 Samson wins the remaining $\mathrm{FO}^2_{m-1,n-1}$ game starting with a move on w'. □

Lemma 1.2.4. Let $m, n \in \mathbb{N}^+$ with $m \leq n$, and let $r \in R_{m,n}$. There is a formula $\psi_r \in \mathrm{FO}^2_{m,n}[<]$ such that for all $w \in \Sigma^\star$ and for all $i \in |w|$, $(w, i) \models \psi_r$ iff $i = r(w)$.

Proof. As in the proof of Lemma 1.2.3, it suffices to show that Samson wins the game $\mathrm{FO}^2_{m,n}(w, w')$ where initially $x^w = r(w)$ and $x^{w'} \neq r(w')$. Depending on whether r is defined over w', we use the strategies from Lemma 1.2.2 or Lemma 1.2.3. □

Theorem 1.2.5 (structure of $\mathrm{FO}^2_{m,n}[<]$)**.** Let w and w' be finite words, and let $m, n \in \mathbb{N}$ with $m \leq n$. The following two conditions are equivalent.

- (i) (a) $R_{m,n}(w) = R_{m,n}(w')$, and,
  - (b) for all $r \in R^\star_{m,n}(w)$ and for all $r' \in R^\star_{m-1,n-1}(w)$, we have $\mathrm{ord}(r(w), r'(w)) = \mathrm{ord}(r(w'), r'(w'))$, and,
  - (c) for all $r \in R^\star_{m,n}(w)$ and $r' \in R^\star_{m,n-1}(w)$ such that r and r' end with different directions, $\mathrm{ord}(r(w), r'(w)) = \mathrm{ord}(r(w'), r'(w'))$
- (ii) $w \equiv^2_{m,n} w'$

Just as before with Theorem 1.1.8, instead of proving Theorem 1.2.5 directly, we prove a more general version that applies to words with two interpreted variables. The statement of the general version is asymmetric with respect to the roles of the two structures w and w'. This is necessary because of the correspondence between quantifier alternations (i.e. alternations between w and w' in the game) and alternations of directions in the rankers. This asymmetry already affected the statement of Lemma 1.2.2, where Samson's winning strategy starts with a move on the specified structure. In fact, as the proof of the following theorem shows, he does not have a winning strategy that starts with a move on the other structure. We remark that conditions (i)(a) through (i)(e) of the general theorem are completely symmetric with respect to the roles of w and w', and only conditions (i)(f) and (ii) are asymmetric. Theorem 1.2.5 follows directly from the general theorem, since here $i_1 = i_2 = j_1 = j_2 = 1$, thus conditions (i)(e) and (i)(f) are trivially true, and the equivalence holds with the roles of w and w' reversed as well.

Theorem 1.2.6. Let w and w' be finite words, let $i_1, i_2 \in |w|$, let $j_1, j_2 \in |w'|$, and let $m, n \in \mathbb{N}$ with $m \leq n$. The following two conditions are equivalent.

(i) (a) $R_{m,n}(w) = R_{m,n}(w')$, and,

(b) for all $r \in R^\star_{m,n}(w)$ and for all $r' \in R^\star_{m-1,n-1}(w)$, we have $\mathrm{ord}(r(w), r'(w)) = \mathrm{ord}(r(w'), r'(w'))$, and,

(c) for all $r \in R^\star_{m,n}(w)$ and $r' \in R^\star_{m,n-1}(w)$ such that r and r' end with different directions, $\mathrm{ord}(r(w), r'(w)) = \mathrm{ord}(r(w'), r'(w'))$

(d) $(w, i_1, i_2) \equiv^2_0 (w', j_1, j_2)$, and,

(e) for all $r \in R^\star_{m-1,n}(w)$, $\mathrm{ord}(r(w), i_1) = \mathrm{ord}(r(w'), j_1)$ and $\mathrm{ord}(r(w), i_2) = \mathrm{ord}(r(w'), j_2)$, and,

(f) for all $r \in R^\star_{m,n}(w)$, and $(i, j) \in \{(i_1, j_1), (i_2, j_2)\}$,

(f_1) if r ends on $\triangleright$ and $r(w) = i$, then $r(w') \leq j$

(f_2) if r ends on $\triangleright$ and $r(w) < i$, then $r(w') < j$

(f_3) if r ends on $\triangleleft$ and $r(w) = i$, then $r(w') \geq j$

(f_4) if r ends on $\triangleleft$ and $r(w) > i$, then $r(w') > j$

(ii) Delilah wins the game $\mathrm{FO}^2_{m,n}[<]((w, i_1, i_2), (w', j_1, j_2))$ if Samson starts with a move on (w, i_1, i_2).

Proof. As in the proof of Theorem 1.1.8, we use induction on n. For $n = 0$, condition (i)(d) just by itself is equivalent to (ii), and all other conditions of (i) are vacuous. For $n \geq 1$, we we first show "$\neg$ (i) $\Rightarrow$ $\neg$ (ii)".

Suppose that (i) holds for (m, n), but fails for $(m, n+1)$. If (i)(d) does not hold then Samson wins immediately. If (i)(e) does not hold for $(m, n+1)$, then by Lemma 1.2.2, Samson wins the $(m, n+1)$-game on (w, w'), starting with a move on either w or w'. If Samson can start with a move on w, we have established that (ii) is false. Otherwise, we reverse the roles of w and w', and observe that condition (i)(e) still remains the same. Thus, even if Samson needs to start with a move on w', he still has a winning strategy, and (ii) does not hold for $(m, n+1)$. If (i)(f) does not hold for $(m, n+1)$, then again by using Lemma 1.2.2, Samson wins the $(m, n+1)$-game on (w, w') starting with a move on w.

If one of (i)(a), (i)(b) or (i)(c) fail, then we show that Samson has a winning strategy for the game $\mathrm{FO}^2_{m,n+1}(w, w')$. We observe that it does not matter what structure Samson chooses for his first move, since all of (i)(a), (i)(b) and (i)(c) are completely symmetric with respect to the roles of w and w'. Thus if Samson's winning strategy starts with a move on w', we can reverse the roles of w and w' and get a winning strategy starting with move on w. One of the following cases applies.

(1) There is a ranker $r \in R_{m,n+1}$ that is defined over one structure but not over the other.

This first case applies if (a) fails for $(m, n+1)$. If condition (2) fails for $(m, n+1)$, then there are two n-rankers for which it fails, or an $(n+1)$-ranker and an n-ranker. This leads to the following two cases.

(2) There are two rankers $r \in R_{m,n}(w)$ and $r' \in R_{m-1,n}(w)$ that disagree on their order, i.e. $\mathrm{ord}(r(w), r'(w)) \neq \mathrm{ord}(r(w'), r'(w'))$.

(3) There are two rankers $r \in R_{m,n+1}(w)$ and $r' \in R_{m-1,n}(w)$ that disagree on their order.

In a similar fashion, we obtain the remaining two cases if condition (3) fails for $(m, n+1)$.

(4) There are rankers $r, r' \in R_{m,n}(w)$ that end on different directions and disagree on their order.

(5) There are rankers $r \in R_{m,n+1}(w)$ and $r' \in R_{m,n}(w)$ that end on different directions and disagree on their order.

We look at the cases (2) and (4) first, then deal with case (1) assuming that cases (2) and (4) do not apply, and finally look at cases (3) and (5).

For case (2), we assume that $r(w) \leq r'(w)$, as illustrated in Figure 1.8. The situation for $r(w) \geq r'(w)$ is completely symmetric. Depending on the last boundary position of r, one of the following two subcases applies.

If r ends with $\triangleright$, then Samson places x on $r(w)$ in his first move. If Delilah replies with a position to the left of $r'(w')$ or equal to $r'(w')$, then $x^{w'} < r(w')$. Thus we can apply Lemma 1.2.2 to get a winning strategy for Samson in the remaining $\mathrm{FO}^2_{m,n}$ game that starts with a move on w. If Delilah replies with a position to the right of $r'(w')$, Samson has a winning strategy for the remaining $\mathrm{FO}^2_{m-1,n}$ game. Thus we have a winning strategy for Samson in the $\mathrm{FO}^2_{m,n+1}$ game.

If on the other hand r ends with $\triangleleft$, then Samson places x on $r(w')$ in his first move. If Delilah replies with a position to the right of $r'(w)$, or equal to $r'(w)$, then

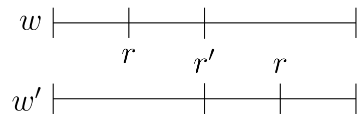

Figure 1.8. Proof of Theorem 1.2.6: r and r' appear in different order.

as above we get a winning strategy for Samson in the remaining $\mathrm{FO}^2_{m,n}$ game that starts with a move on w'. Otherwise we get a winning strategy for Samson with only $m-1$ alternations for the remaining game. Thus again he has a winning strategy for the $\mathrm{FO}^2_{m,n+1}$ game.

For case (4), Samson's winning strategy is very similar to the previous case. If $r(w) \leq r'(w)$ and r ends with $\triangleright$, then Samson places x on $r(w)$ in his first move. If Delilah replies with a position to the right of $r(w)$, then Samson's winning strategy is as above. Otherwise x is on different sides of r' and Samson has a winning strategy for the remaining $\mathrm{FO}^2_{m,n}$ game that starts with a move on w. All together, he has a winning strategy for the $\mathrm{FO}^2_{m,n+1}$ game. The remaining three cases (ordering of $r(w)$ and $r'(w)$ and ending direction of r) work in the same way.

Similar to what we did in the proof of Theorem 1.1.8, we can reduce the remaining cases to an easier situation where a certain segment contains a certain letter in one structure, but not in the other structure, and then apply Lemma 1.2.2 to obtain a winning strategy for Samson.

To deal with case (1), we assume that the previous two cases, (2) and (4), do not apply. Without loss of generality, say that the $(m, n+1)$-ranker r is defined over w but not over w'. Let $\mathtt{a} := w_{r(w)}$ be the letter in w at position $r(w)$. We define the following sets of rankers.

$$R_\ell := \{s \in R^\star_{m\triangleright,n}(w) \mid s(w) < r(w)\}$$

$$R_r := \{s \in R^\star_{m\triangleleft,n}(w) \mid s(w) > r(w)\}$$

Notice that all rankers from R_ℓ appear to the left of all rankers from R_r in w. From the inductive hypothesis, and from the fact that both cases (2) and (4) do not apply, it follows that over w', all rankers from R_ℓ appear to the left of all rankers from R_r as well. However, the rankers from R_ℓ and R_r by themselves do not necessarily appear in the same order in both structures. We look at the ordering of these rankers in w', and let λ be the rightmost ranker from R_ℓ and ρ be the leftmost ranker from R_r. By construction, we have $\lambda(w) < r(w) < \rho(w)$, so the segment (λ, ρ) in w contains the letter $\mathtt{a}$. Let r_n be the n-prefix-ranker of r, and observe that r_n is defined on both structures and that r_n is contained in either R_ℓ or R_r. Because r is not defined on w', the letter $\mathtt{a}$ does not occur in w' either to the right of r_n if $r_n \in R_\ell$, or to the left of r_n if $r_n \in R_r$. Thus the segment (λ, ρ) does not contain the letter $\mathtt{a}$ in w'.

Now we know that $\mathtt{a}$ occurs in the segment (λ, ρ) in w but not in w', and thus we have established the situation illustrated in Figure 1.9. Samson places his first pebble on an $\mathtt{a}$ within this section of w, and Delilah has to reply with a position outside of this section. No matter what side of the segment she chooses, with Lemma 1.2.2 Samson has a winning strategy for the remaining game and thus wins the $\mathrm{FO}^2_{m,n+1}$ game. In cases (3) and (5), we again assume that cases (2) and (4) do not apply,

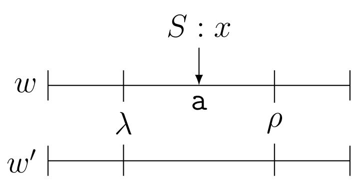

Figure 1.9. Proof of Theorem 1.2.6: A letter occurs between rankers r, r' in w but not in w'.

and we look at the same sets of rankers, R_ℓ and R_r, and at r_n, the n-prefix-ranker of r. We assume that $r(w) \le r'(w)$ and that r ends with $\triangleright$, all three other cases are completely symmetric. Notice that r_n is an $(m-1, n)$-ranker, or an (m, n)-ranker that ends with $\triangleright$. Thus both structures agree on the ordering of r_n and r'. The relative positions of all these rankers are illustrated in Figure 1.10. As above, let λ be the rightmost ranker from R_ℓ and let ρ be the leftmost ranker from R_r, with respect to the ordering of these rankers on w'. Again we know that $\lambda(w) < r(w) < \rho(w)$ and therefore the segment (λ, ρ) of w contains an $\mathtt{a}$. Notice that $r_n \in R_\ell$ and $r' \in R_r$, thus $r_n(w') \leq \lambda(w') < \rho(w') \leq r'(w')$. Thus the segment (λ, ρ) does not contain the letter $\mathtt{a}$ in w', providing Samson with a winning strategy as argued above.

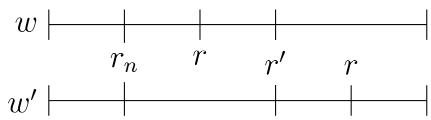

Figure 1.10. Proof of Theorem 1.2.6: Ranker positions, case (4).

To prove "(i) $\Rightarrow$ (ii)", we assume that the theorem holds for n, and that (i) holds for $(m, n+1)$, and we present a winning strategy for Delilah in the game $\mathrm{FO}^2_{m,n+1}(w, w')$ where Samson starts with a move on w.

If Samson places x on a ranker $r \in R^\star_{m-1,n}(w)$, then Delilah replies by placing x on the same ranker on w'. Since (i)(b) holds for $(m, n+1)$, this establishes (i)(e) and (i)(f) for (m, n). It also establishes (i)(e) and (i)(f) for $(m-1, n)$ with reversed roles of w and w'. Thus we can apply the inductive hypothesis to get a winning strategy for Delilah in the remaining game.

If $x^w = y^w$ after Samson's first move, then Delilah replies with $x^{w'} = y^{w'}$. We use the inductive hypothesis to argue that Delilah wins the remaining n-move game, no matter what structure Samson chooses for his next move. If he chooses to play on w, then the remaining game is an (m, n)-game. Since in the first move Delilah set $x^{w'} = y^{w'}$, we have (i)(e) and (i)(f) for (m, n), and thus the inductive hypothesis applies and Delilah wins the remaining game. On the other hand, if Samson chooses to play on w' for the next move, the remaining game is an $(m-1, n)$-game, since he started with a move on w. Because Delilah set $x^{w'} = y^{w'}$ in the first move, (i)(e) for $(m, n+1)$ implies both (i)(e) and (i)(f) for $(m-1, n)$ with reversed roles of w and w'.

Thus we can again use the inductive hypothesis to get a winning strategy for Delilah in the remaining game.

Otherwise we assume that $x^w < y^w$ after Samson's first move, the case for $x^w > y^w$ is completely symmetric. We look at the following two sets of rankers.

$$R_\ell := \{r \in R^\star_{m\triangleright,n}(w) \mid r(w) < x^w\}$$
$$R_r := \{r \in R^\star_{m\triangleleft,n}(w) \mid r(w) > x^w\}$$

On w, all rankers from R_ℓ occur to the left of all rankers from R_r. Since (i)(c) holds for $(m, n+1)$, this is also true for the positions of these rankers on w'. Let $\mathtt{a}$ be the letter Samson places his pebble on. To establish both (i)(e) and (i)(f) for (m, n), Delilah needs to find an $\mathtt{a}$ in w' that is to the right of all rankers from R_ℓ and to the left of all rankers from R_r. We define

$$R^0_\ell := \{r \in R^\star_{m\triangleright,n}(w) - R^\star_{m-1,n}(w) \mid r(w) = x^w\}$$
$$R^0_r := \{r \in R^\star_{m\triangleleft,n}(w) - R^\star_{m-1,n}(w) \mid r(w) = x^w\}$$
$$R'_\ell := \{r\triangleright_{\mathtt{a}} \mid r \in R_\ell\} \cup R^0_\ell \quad ,$$

and have Delilah place her pebble $x^{w'}$ on the rightmost ranker from R'_ℓ on w'. This position of course is labeled with an $\mathtt{a}$. Since on w all rankers from R'_ℓ occur to the left of or at x^w, all of them occur strictly to the left of y^w. Since all rankers in R'_ℓ are from $R^\star_{m-1,n+1}(w)$ or $R^\star_{m\triangleright,n+1}(w)$, we can apply (i)(e) and (i)(f$_2$), and we see that all of these rankers also appear to the left of $y^{w'}$. Therefore we have $x^{w'} < y^{w'}$, which makes sure that Delilah does not lose in this move, and also establishes (i)(d).

To complete the inductive step, we need to argue that Delilah's move also establishes (i)(e) and (i)(f), both for (m, n), and for $(m-1, n)$ with reversed roles of w and w'. Then, using the inductive hypothesis, Delilah has a winning strategy for the remaining game, no matter what side Samson chooses for his next move.

We observe that all rankers from R'_ℓ appear to the right of the rankers from R_r. This is true by definition on w, and holds for w' because (i)(b) and (i)(c) hold for $(m, n+1)$. Since Delilah placed $x^{w'}$ on a ranker from R'_ℓ, we have (i)(e), (i)(f_2) and (i)(f_4) for (m, n) for all all rankers from R_r. And since Delilah placed $x^{w'}$ on the rightmost of the rankers from R'_ℓ, we know that all rankers from R_ℓ appear to the left of $x^{w'}$, just as they do on w. Thus we have (i)(e), (i)(f_2) and (i)(f_4) for the rankers from R_ℓ as well, and therefore for all rankers mentioned in those conditions.

All rankers from $R^{\star}_{m\triangleright,n}$ that appear at x^w are in R^0_ℓ, since we already dealt with the case where x^w does appear at a ranker from $R^{\star}_{m-1,n}$. Since Delilah chose $x^{w'}$ as the rightmost ranker from R'_ℓ, all of these rankers appear to the left of or at $x^{w'}$, and we have established (i)(f_1) for (m, n). For condition (i)(f_3), we need to argue about R^0_r. From (i)(b) and (i)(c) for $(m, n+1)$, we know that all rankers from R^0_r appear to the right of or at the same position as the rankers from R'_ℓ on w', just as they do on w. Thus (i)(f_3) holds as well.

Now that we have established (i) for (m, n), we use the inductive hypothesis to get a winning strategy for Delilah for the remaining game if Samson's next move is on w. For the case where his next move is on w', we only need to establish (i) for $(m-1, n)$, but with reversed roles of w and w'. Reversing the roles of the two structures only affects condition (i)(f), and (i)(f) for $(m-1, n)$ follows immediately from (i)(e) for (m, n). Thus Delilah also wins the remaining game if Samson's next move is on w'. $\square$

Using Theorem 1.2.5, we show that for any fixed alphabet Σ, at most $|\Sigma| + 1$ alternations are useful. Intuitively, each boundary position in a ranker says that a certain letter does not occur in some part of a word. Alternations are only useful if they visit one of these previous parts again. Once we visited one part of a word $|\Sigma|$ times, this part cannot contain any more letters and thus is empty.

Theorem 1.2.7. Let Σ be a finite alphabet, let $w, w' \in \Sigma^\star$ and $n \in \mathbb{N}$. If $w \equiv^2_{|\Sigma|+1,n} w'$, then $w \equiv^2_n w'$.

Proof. Suppose for the sake of a contradiction that $w \equiv^2_{|\Sigma|+1,n} w'$ and $w \not\equiv^2_n w'$. Thus, using Theorem 1.2.5, w and w' agree on the definedness of all $(|\Sigma|+1, n)$-rankers, and on their order with respect to all $(|\Sigma|, n-1)$-rankers and some $(|\Sigma|+1, n-1)$-rankers. But since $w \not\equiv^2_n w'$, w and w' need to disagree on the properties of some other ranker. Let $r := (p_1, \ldots, p_t)$ with $t \in \mathbb{N}$ be the shortest such ranker. We know that r has more than $|\Sigma|$ blocks of alternating directions, say r is an m-alternation ranker for some $m > |\Sigma|$. Let $1 \leq k_1, \ldots, k_m \leq t$ be the indices of the boundary positions at the end of each block, i.e. where p_{k_i}, $1 \leq i < m$ points to a different direction than p_{k_i+1}. For the last of those indices we have $k_m = t$.

We look at the prefix rankers of r up to the end of each alternating block, $r_{k_i} := (p_1, \ldots, p_{k_i})$, and the intervals defined by these prefix rankers. We set $I_0(w) := |w|$, $r_0(w) := 0$ if p_1 points to the right, and $r_0(w) := \|w\| + 1$ if p_1 points to the left. For all $i \in [1, m]$, let

$$I_i(w) := \begin{cases} [r_{k_i-1}(w) + 1, r_{k_i}(w) - 1] & \text{if } p_{k_i} \text{ points to the right} \\ [r_{k_i}(w) + 1, r_{k_i-1}(w) - 1] & \text{if } p_{k_i} \text{ points to the left} \end{cases} .$$

Notice that by definition the letter mentioned in p_{k_i} does not occur in the interval I_i.

Suppose that for all $i \in [1, m]$ we have $r_{k_i}(w) \in I_{i-1}(w)$. Then the letter mentioned in p_{k_i} has to occur in the interval $I_{i-1}(w)$ of w, but the interval $I_{|\Sigma|}(w)$ of w cannot contain any of the $|\Sigma|$ distinct letters. Therefore $r_{k_{|\Sigma|+1}} \notin I_{|\Sigma|}$ and we have a contradiction.

Otherwise there is an $i \in [1, m]$ such that $r_{k_i}(w) \notin I_{i-1}(w)$. We will construct a ranker r' that is shorter than r, does not have more alternations than r and occurs at exactly the same position as r in both w and w'. The main idea for this construction is that if $r_{k_i}(w) \notin I_{i-1}(w)$, then it is not useful to enter this interval at all. By our assumption, w and w' disagree on some property of the ranker r, and thus on some property of the shorter ranker r'. This contradicts our assumption that r was the shortest such ranker.

Now we show how to construct a shorter ranker r' that occurs at the same position as r. We assume without loss of generality that p_{k_i} points to the left. In this case we have $r_{k_i}(w) \notin I_{i-1}(w) = [r_{k_{i-1}-1}(w) + 1, r_{k_{i-1}}(w) - 1]$. We look at the relative positions of the rankers $r_{k_{i-1}+1}, \ldots, r_{k_i}$ with respect to the ranker $r_{k_{i-1}-1}$. We know that $r_{k_i}(w) \leq r_{k_{i-1}-1}(w)$, and we are interested in the right-most of the rankers $r_{k_{i-1}+1}, \ldots, r_{k_i}$ that is still outside of the interval $I_{i-1}(w)$. Let r_j be this ranker. Thus we have

$$r_{k_i}(w) < \ldots < r_j(w) \leq r_{k_{i-1}-1}(w) < r_{j-1}(w) < \ldots < r_{k_{i-1}+1}(w) < r_{k_{i-1}}(w) \quad .$$

We know that $w \equiv^2_{|\Sigma|+1,n} w'$, thus by Theorem 1.2.5, these rankers occur in exactly the same order in w'. Now we set $s := (r_{k_{i-1}-1}, p_j, \ldots, p_{k_i})$. Because w and w' agree on the ordering of the relevant rankers, we have $s(w) = r_{k_i}(w)$ and $s(w') = r_{k_i}(w')$. Therefore we have reduced the size of a prefix of r without increasing the number of alternations, and thus have a shorter ranker r' that occurs at the same position as r in both structures. $\square$

In order to prove that the alternation hierarchy for FO^2 is strict, we define example languages that can be separated by a formula of a given alternation depth m, but that cannot be separated by any formula of lower alternation depth. As Theorem 1.2.7 shows, we need to increase the size of the alphabet with increasing alternation depth. We inductively define the example words $w_{m,n}$ and $w'_{m,n}$ and the example languages K_m and L_m over finite alphabets $\Sigma_m = \{a_0, \ldots, a_{m-1}\}$. Here i, m and n are positive integers.

$$
\begin{aligned}
w_{1,n} &:= a_0 & w'_{1,n} &:= \varepsilon \\
w_{2,n} &:= a_0(a_1a_0)^{2n} & w'_{2,n} &:= (a_1a_0)^{2n} \\
w_{2i+1,n} &:= (a_0 \dots a_{2i})^n \; w_{2i,n} & w'_{2i+1,n} &:= (a_0 \dots a_{2i})^n \; w'_{2i,n} \\
w_{2i+2,n} &:= w_{2i+1,n} \; (a_{2i+1} \dots a_0)^n & w'_{2i+2,n} &:= w'_{2i+1,n} \; (a_{2i+1} \dots a_0)^n
\end{aligned}
$$

Notice that $w_{m,n}$ and $w'_{m,n}$ are almost identical – if we delete only one a_0 from $w_{m,n}$, we get $w'_{m,n}$. Finally, we set $K_m := \bigcup_{n\geq 1}\{w_{m,n}\}$ and $L_m := \bigcup_{n\geq 1}\{w'_{m,n}\}$.

Definition 1.2.8. A formula φ *separates* two languages $K, L \subseteq \Sigma^\star$ if for all $w \in K$ we have $w \models \varphi$ and for all $w \in L$ we have $w \not\models \varphi$ or vice versa.

Lemma 1.2.9. For all $m \in \mathbb{N}$, there is a formula $\varphi_m \in \mathrm{FO}^2[<]\text{–ALT}\,[\mathrm{m}]$ that separates K_m and L_m.

Proof. For $m = 1$, we can easily separate $K_1 = \{a_0\}$ and $L_1 = \{\varepsilon\}$ with the formula $\exists x\,(x = x)$. For all larger m, we show that the two languages K_m and L_m differ on the ordering of two $(m-1)$-alternation rankers. Then by Theorem 1.2.5 there is an $\mathrm{FO}^2_{m,m}[<]$ formula that separates K_m and L_m. We inductively define the rankers

$$
\begin{aligned}
r_2 &:= \triangleright_{a_0} & s_2 &:= \triangleright_{a_1} \\
r_{2i+1} &:= \triangleleft_{a_{2i}} r_{2i} & s_{2i+1} &:= \triangleleft_{a_{2i}} s_{2i} \\
r_{2i+2} &:= \triangleright_{a_{2i+1}} r_{2i+1} & s_{2i+2} &:= \triangleright_{a_{2i+1}} s_{2i+1} \quad .
\end{aligned}
$$

For $m = 2$, it is easy to see that $r_2(w_{2,n}) < s_2(w_{2,n})$, but $r_2(w'_{2,n}) > s_2(w'_{2,n})$. For $m > 2$, these rankers disagree on their order as well. To prove this, we prove the following two equalities.

$$
r_{2i+2}(w_{2i+2,n}) = r_{2i+1}(w_{2i+1,n}) = (2i+1)n + r_{2i}(w_{2i,n})
$$

To prove this, we first use the definitions above and write

$$r_{2i+2}(w_{2i+2,n}) = (\triangleright_{a_{2i+1}} r_{2i+1})(w_{2i+1,n}\ (a_{2i+1} \ldots a_0)^n) \quad .$$

The letter a_{2i+1} does not occur in the word $w_{2i+1,n}$, and thus $\triangleright_{a_{2i+1}}(w_{2i+2,n})$ points to the first position in $w_{2i+2,n}$ right after the copy of $w_{2i+1,n}$. We observe that r_{2i+1} starts with $\triangleleft$, and that r_{2i+1} is defined on $w_{2i+1,n}$. Thus the evaluation of the remainder of r_{2i+2} on $w_{2i+2,n}$ never leaves the copy of $w_{2i+1,n}$, and we have

$$r_{2i+2}(w_{2i+2,n}) = r_{2i+1}(w_{2i+1,n}) \quad .$$

For the second part of the equality, we have

$$r_{2i+1}(w_{2i+1,n}) = (\triangleleft_{a_{2i}} r_{2i})((a_0 \ldots a_{2i})^n\ w_{2i,n}) \quad .$$

As above, the letter a_{2i} does not occur in the word $w_{2i,n}$, and thus $\triangleleft_{a_{2i}}(w_{2i+1,n})$ points to the position in $w_{2i+1,n}$ right before the copy of $w_{2i,n}$. The ranker r_{2i} starts with $\triangleright$, and r_{2i} is defined on $w_{2i,n}$. Thus, just as above, the evaluation of the remainder of r_{2i+1} on $w_{2i+1,n}$ never leaves the copy of $w_{2i,n}$, and we have

$$r_{2i+1}(w_{2i+1,n}) = (2i+1)n + r_{2i}(w_{2i,n}) \quad .$$

Exactly the same holds for the other rankers $(s_2, \ldots)$ and words $(w'_{2,n}, \ldots)$. We have

$$\begin{aligned}
r_{2i+2}(w_{2i+2,n}) &= r_{2i+1}(w_{2i+1,n}) = (2i+1)n + r_{2i}(w_{2i,n}) \\
s_{2i+2}(w_{2i+2,n}) &= s_{2i+1}(w_{2i+1,n}) = (2i+1)n + s_{2i}(w_{2i,n}) \\
r_{2i+2}(w'_{2i+2,n}) &= r_{2i+1}(w'_{2i+1,n}) = (2i+1)n + r_{2i}(w'_{2i,n}) \\
s_{2i+2}(w'_{2i+2,n}) &= s_{2i+1}(w'_{2i+1,n}) = (2i+1)n + s_{2i}(w'_{2i,n}) \quad .
\end{aligned}$$

Now an easy inductive argument, based on the two equalities we just proved, shows that the rankers disagree on their order. Therefore condition (i)(b) of Theorem 1.2.5 fails for any pair of words, and there is a formula in $\mathrm{FO}^2_{m,m}[<]$ that separates K_m and L_m. □

Lemma 1.2.10. For $m \in \mathbb{N}$, $m \geq 1$, and all $n \in \mathbb{N}$, we have $w_{m,n} \equiv^2_{m-1,n} w'_{m,n}$.

Proof. Because we do not have constants, there are no quantifier-free sentences. Thus $\mathrm{FO}^2_{0,n}[<]$ does not contain any formulas and the statement holds trivially for $m = 1$.

For $m \geq 2$ and any $n \geq m$, we claim that exactly the same $(m-1,n)$-rankers are defined over $w_{m,n}$ and $w'_{m,n}$, and that all $(m-1,n)$-rankers appear in the same order with respect to all $(m-2,n-1)$-rankers and all $(m-1,n-1)$-rankers that end on a different direction. Once we established this claim, the lemma follows immediately from Theorem 1.2.5. We already observed that $w_{m,n}$ and $w'_{m,n}$ are almost identical. The only difference between the two words is that $w_{m,n}$ contains the letter a_0 in the middle whereas $w'_{m,n}$ does not. Thus we only have to consider rankers that are affected by this middle a_0.

We claim that any ranker that points to the middle a_0 of $w_{m,n}$ requires at least $m-1$ alternations. Furthermore, we claim that any such ranker needs to start with $\triangleright$ for even m and with $\triangleleft$ for odd m. We prove this by induction on m.

For $m = 2$ we have $w_{2,n} = a_0(a_1a_0)^n$. Any n-ranker that starts with $\triangleleft$ cannot reach the first a_0, thus we need a ranker that starts with $\triangleright$.

For odd $m > 2$ we have $w_{m,n} = (a_0 \ldots a_{m-1})^n w_{m-1,n}$. Any n-ranker that starts with $\triangleright$ cannot leave the first block of $n \cdot m$ symbols of this word and thus not reach the middle a_0. Therefore we need to start with $\triangleleft$, and in fact use $\triangleleft_{a_{m-1}}$ at some point, because we would not be able to leave the last section of $w_{m-1,n}$ otherwise. But with $\triangleleft_{a_{m-1}}$ we move past all of $w_{m-1,n}$, and we need one alternation to turn around again.

By induction, we need at least $m-2$ alternations within $w_{m-1,n}$, and thus $m-1$ alternations total.

The argument for even m is completely symmetric. Thus we showed that we need at least $m-1$ alternation blocks to point to the middle a_0. Furthermore, we showed that if we have exactly $m-1$ alternation blocks, then the last of these blocks uses $\triangleright$. Therefore we only need to consider $(m-1)$-alternation rankers that end on $\triangleright$ and pass through the middle a_0. It is easy to see that all of these rankers agree on their ordering with respect to all other $(m-2)$-alternation rankers, and with respect to all $(m-1)$-alternation rankers that end on $\triangleleft$.

To summarize, we showed that $w_{m,n}$ and $w'_{m,n}$ satisfy condition (i) from Theorem 1.2.5 for $m-1$ alternations. Thus these two words agree on all $\mathrm{FO}^2_{m-1,n}[<]$ formulas. $\square$

Theorem 1.2.11 (alternation hierarchy for $\mathrm{FO}^2[<]$)**.** For any positive integer m, there is a $\varphi_m \in \mathrm{FO}^2\text{–ALT}\,[m]\,[<]$ and there are two languages K_m, L_m such that φ_m separates K_m and L_m, but no $\psi \in \mathrm{FO}^2\text{–ALT}\,[m-1]\,[<]$ separates K_m and L_m.

Proof. The theorem immediately follows from Lemma 1.2.9 and Lemma 1.2.10. $\square$

1.3 Extension to $\mathbf{FO}^2[<, \mathrm{Suc}]$

We extend our definitions of boundary positions and rankers from Section 1.1 to include the substrings of a given length that occur immediately before and after the position of the ranker.

Definition 1.3.1. A (k,ℓ)-*neighborhood boundary position* denotes the first or last occurrence of a substring in a word. More precisely, a (k,ℓ)-neighborhood boundary position is of the form $d_{(s,\mathtt{a},t)}$ with $d \in \{\triangleright, \triangleleft\}$, $s \in \Sigma^k$, $\mathtt{a} \in \Sigma$ and $t \in \Sigma^\ell$. The interpretation of a (k,ℓ)-neighborhood boundary position $p = d_{(s,\mathtt{a},t)}$ on a word $w = w_1 \dots w_{\|w\|}$ is defined as follows.

$$p(w) := \begin{cases} \min\{i \in [k+1, \|w\| - \ell] \mid w_{i-k} \dots w_{i+\ell} = s\,\mathtt{a}\,t\} & \text{if } d = \triangleright \\ \max\{i \in [k+1, \|w\| - \ell] \mid w_{i-k} \dots w_{i+\ell} = s\,\mathtt{a}\,t\} & \text{if } d = \triangleleft \end{cases}$$

Notice that $p(w)$ is undefined if the sequence $s\mathtt{a}t$ does not occur in w. A (k,ℓ)-neighborhood boundary position can also be specified with respect to a position $q \in |w|$.

$$p(w,q) := \begin{cases} \min\{i \in [\max\{q+1, k+1\}, \|w\| - \ell] \mid w_{i-k} \dots w_{i+\ell} = s\,\mathtt{a}\,t\} & \text{if } d = \triangleright \\ \max\{i \in [k+1, \min\{q-1, \|w\| - \ell\}] \mid w_{i-k} \dots w_{i+\ell} = s\,\mathtt{a}\,t\} & \text{if } d = \triangleleft \end{cases}$$

Observe that $(0,0)$-neighborhood boundary positions are identical to the boundary positions from Definition 1.1.1. As before in the case without successor, we build rankers out of these boundary positions. The size of the boundary position neighborhoods grows linearly from the first boundary position to the last one, reflecting the remaining quantifier depth for successor moves at those positions.

Definition 1.3.2. An *n-successor-ranker* r is a sequence of n neighborhood boundary positions, $r = (p_1, \dots, p_n)$, where p_i is a (k_i, ℓ_i)-neighborhood boundary position and $k_i, \ell_i \in [0, i-1]$. The interpretation of an n-successor-ranker r on a word w is defined as follows.

$$r(w) := \begin{cases} p_1(w) & \text{if } r = (p_1) \\ \text{undefined} & \text{if } (p_1, \dots, p_{n-1})(w) \text{ is undefined} \\ p_n(w, (p_1, \dots, p_{n-1})(w)) & \text{otherwise} \end{cases}$$

We denote the set of all n-successor-rankers that are defined over a word w by $SR_n(w)$, and set $SR_n^\star(w) := \bigcup_{i \in [1,n]} SR_i(w)$.

Because we now have the additional atomic relation Suc, we need to extend our definition of order type as well.

Definition 1.3.3. Let $i, j \in \mathbb{N}$. The *successor order type* of i and j is defined as

$$\operatorname{ord}_{\mathrm{S}}(i,j) := \begin{cases} \ll & \text{if } i < j-1 \\ -1 & \text{if } i = j-1 \\ = & \text{if } i = j \\ +1 & \text{if } i = j+1 \\ \gg & \text{if } i > j+1 \end{cases} .$$

With this new definition of n-successor-rankers, our proofs for Lemmas 1.1.5, 1.1.6, 1.1.7 and Theorem 1.1.8 go through with only minor modifications. Instead of working through all the details again, we simply point out the differences.

First we notice that 1-successor-rankers are simply 1-rankers, so the base case of all inductions remains unchanged. In the proofs of Lemmas 1.1.5, 1.1.6 and 1.1.7, and in the proof of "(ii) $\Rightarrow$ (i)" from Theorem 1.1.8, we argued that Delilah cannot reply with a position in a given section because it does not contain a certain ranker and therefore it does not contain the symbol used to define this ranker. Now we need to know more – we need to show that Delilah cannot reply with a certain letter in a given section that is surrounded by a specified neighborhood, given that this section does not contain the corresponding successor-ranker. Whenever Samson's winning strategy depends on the fact that an n-successor-ranker does not occur in a given section, he has $n-1$ additional moves left. So if Delilah does not reply with a position with the same letter and the same neighborhood, Samson can point out a difference in the neighborhood with at most $(n-1)$ additional moves.

For the other direction of Theorem 1.1.8, we need to make sure that Delilah can reply with a position that is contained in the correct interval, has the same symbol and is surrounded by the same neighborhood. Where we previously defined the n-

ranker $s := (\lambda, \triangleright_{\mathtt{a}})$ or $s := (\rho, \triangleleft_{\mathtt{a}})$, we now include the $(n-1)$-neighborhood of the respective positions chosen by Samson. Thus we make sure that Samson cannot point out a difference in the two words, and Delilah still has a winning strategy. Thus we have the following three theorems for $\mathrm{FO}^2[<, \mathrm{Suc}]$.

Theorem 1.3.4 (structure of $\mathrm{FO}^2_n[<, \mathrm{Suc}]$)**.** Let w and w' be finite words, and let $n \in \mathbb{N}$. The following two conditions are equivalent.

(i) (a) $SR_n(w) = SR_n(w')$, and,

(b) for all $r \in SR^\star_n(w)$ and for all $r' \in SR^\star_{n-1}(w)$,

$\mathrm{ord}_{\mathrm{S}}(r(w), r'(w)) = \mathrm{ord}_{\mathrm{S}}(r(w'), r'(w'))$

(ii) $w \equiv^2_n w'$

Theorem 1.3.5 (structure of $\mathrm{FO}^2_{m,n}[<, \mathrm{Suc}]$)**.** Let w and w' be finite words, and let $m, n \in \mathbb{N}$ with $m \leq n$. The following two conditions are equivalent.

(i) (a) $SR_{m,n}(w) = SR_{m,n}(w')$, and,

(b) for all $r \in SR^\star_{m,n}(w)$ and for all $r' \in SR^\star_{m-1,n-1}(w)$,

$\mathrm{ord}_{\mathrm{S}}(r(w), r'(w)) = \mathrm{ord}_{\mathrm{S}}(r(w'), r'(w'))$, and,

(c) for all $r \in SR^\star_{m,n}(w)$ and $r' \in SR^\star_{m,n-1}(w)$ such that r and r' end with different directions, $\mathrm{ord}_{\mathrm{S}}(r(w), r'(w)) = \mathrm{ord}_{\mathrm{S}}(r(w'), r'(w'))$

(ii) $w \equiv^2_{m,n} w'$

Theorem 1.3.6 (alternation hierarchy for $\mathrm{FO}^2[<, \mathrm{Suc}]$)**.** Let m be a positive integer. There is a $\varphi_m \in \mathrm{FO}^2\text{–ALT}\,[m]\,[<, \mathrm{Suc}]$ and there are two languages $K_m, L_m \subseteq \Sigma^\star$ such that φ_m separates K_m and L_m, but there is no $\psi \in \mathrm{FO}^2\text{–ALT}\,[m-1]\,[<, \mathrm{Suc}]$ that separates K_m and L_m.

Proof. We use the same ideas as before in Theorem 1.2.11. We define example languages that now include an extra letter `b` to ensure that the successor predicate is of no use. As before, we inductively construct the words $w_{m,n}$ and $w'_{m,n}$ and use them to define the languages K_m and L_m.

$$
\begin{aligned}
w_{1,n} &:= \mathtt{b}^{2n} a_0 \mathtt{b}^{2n} & w'_{1,n} &:= \mathtt{b}^{2n} \\
w_{2,n} &:= w_{1,n}\ (a_1 \mathtt{b}^{2n} a_0 \mathtt{b}^{2n})^{2n} & w'_{2,n} &:= w'_{1,n}\ (a_1 \mathtt{b}^{2n} a_0 \mathtt{b}^{2n})^{2n} \\
w_{2i+1,n} &:= (\mathtt{b}^{2n} a_0 \mathtt{b}^{2n} \dots \mathtt{b}^{2n} a_{2i})^n\ w_{2i,n} & w'_{2i+1,n} &:= (\mathtt{b}^{2n} a_0 \mathtt{b}^{2n} \dots \mathtt{b}^{2n} a_{2i})^n\ w'_{2i,n} \\
w_{2i+2,n} &:= w_{2i+1,n}\ (a_{2i+1} \mathtt{b}^{2n} \dots a_0 \mathtt{b}^{2n})^n & w'_{2i+2,n} &:= w'_{2i+1,n}\ (a_{2i+1} \mathtt{b}^{2n} \dots a_0 \mathtt{b}^{2n})^n
\end{aligned}
$$

Finally we set $K_m := \bigcup_{n \geq 1} \{w_{m,n}\}$ and $L_m := \bigcup_{n \geq 1} \{w'_{m,n}\}$. Notice that the $\mathtt{b}$s are not necessary to distinguish between the two languages K_m and L_m, and thus the proof of Lemma 1.2.9 goes through unchanged and we have a formula $\varphi_m \in \mathrm{FO}^2\text{–}\mathrm{ALT}\,[m]\,[<, \mathrm{Suc}]$ that separates K_m and L_m. To see that no $\mathrm{FO}^2\text{–}\mathrm{ALT}\,[m-1]\,[<, \mathrm{Suc}]$ formula can separate K_m and L_m, we observe that any $(n-1)$-neighborhood in the words $w_{m,n}$ and $w'_{m,n}$ contains all $\mathtt{b}$s except for at most one letter a_i for some $i \in [0, m-1]$. Thus the proof of Lemma 1.2.10 goes through here as well. $\square$

CHAPTER 2

SATISFIABILITY OF FO^2 ON FINITE WORDS

We investigate the complexity of the satisfiability problem for FO^2 on finite words. While there is a straightforward reduction from boolean satisfiability that shows that satisfiability for $\mathrm{FO}^2[<]$ on finite words is NP-hard, this problem was previously only known to be in NEXP [10], and in NP for the special case where the size of the alphabet is fixed [47]. In this chapter we present a general NP algorithm which, similar to the previously known NEXP algorithm, is based on a small model property.

As defined earlier, finite words are represented as logical structures where the elements of the universe correspond to the positions in the word, and we have unary predicates for each letter in the alphabet. Thus exactly one unary predicate holds at any given position. In a slightly more general model, which we choose to call *finite power words*, any number of unary predicates can hold at any position. In many contexts, this distinction does not matter much, as finite words are finite power words, and we can convert finite power words to finite words by replacing the original alphabet Σ with the new alphabet $\mathcal{P}(\Sigma)$. For example, using this translation, FO^2 on finite words has the same expressive power as FO^2 on finite power words. However, while the satisfiability problem for FO^2 on finite power words is NEXP-complete [10], this result does not hold for finite words.

We present the small model property for $\mathrm{FO}^2[<]$ that our NP satisfiability algorithm is based on in Section 2.1, and conclude this chapter with a complete picture of the complexity of the satisfiability problem for FO^2 on finite words and finite power words in Section 2.2.

2.1 Small Model Property for $\mathrm{FO}^2[<]$ on Finite Words

Theorem 2.1.1. Let $\varphi \in \mathrm{FO}^2[<]$ over the alphabet Σ. If φ is satisfiable, then φ has a model of size $O(|\Sigma| \cdot |\varphi|^3)$.

Before we prove Theorem 2.1.1, we introduce some definitions that will be helpful in case distinctions based on the relative ordering of the interpretations of the two variables x and y in a formula or its subformulas.

Definition 2.1.2. Let $w \in \Sigma^\star$, $a \in \Sigma$ and $S \subseteq |w|$. Then $S^a := \{i \in S \mid w_i = a\}$.

Definition 2.1.3. An *order formula* is one of the three formulas $x < y$, $x = y$ and $x > y$.

Definition 2.1.4. A formula $\varphi \in \mathrm{FO}[<]$ is in *existential negation normal form (ENNF)* if it does not contain any universal quantifiers, and negations only appear in front of unary predicates or existential quantifiers. Negations in front of order formulas are not allowed.

Proposition 2.1.5. Any formula $\varphi \in \mathrm{FO}[<]$ is equivalent to a formula in ENNF of size at most $2|\varphi|$.

Proof. We push all negations inside in the usual way, but not past any existential quantifiers. Negated order formulas are converted to a positive boolean combination of order formulas.

We use induction on the structure of φ to construct an equivalent formula φ' in ENNF. If $\varphi = Q_a(x)$ for some $a \in \Sigma$, or φ is an order formula, then we set $\varphi' = \varphi$. If $\varphi = \psi \vee \xi$, then $\varphi' = \psi' \vee \xi'$. If $\varphi = \exists x\, \psi$, then $\varphi' = \exists x\, \psi'$. Finally if $\varphi = \neg\psi$, we need to look at the structure of ψ:

- If $\psi = Q_a(x)$ for some $a \in \Sigma$, we set $\varphi' = \varphi$.
- If ψ is of the form $x < y$, then φ' is $x = y \vee x > y$.

- If ψ is of the form $x = y$, then φ' is $x < y \vee x > y$.
- If $\psi = \neg\xi$, then $\varphi' = \xi'$.
- If $\psi = \xi \vee \zeta$, then $\varphi' = (\neg\xi)' \wedge (\neg\zeta)'$.
- If $\psi = \exists x\, \psi$, then $\varphi' = \neg\exists x\, \psi'$.

The inductively constructed formula φ' is in ENNF, and it is easy to verify that in each step of the induction $\varphi \equiv \varphi'$. □

Proposition 2.1.6. Let $\varphi \in \mathrm{FO}^2[<]$ in ENNF. Then there are $s \in \mathbb{N}$, a positive boolean formula β in the propositional variables $Z_<, Z_=, Z_>, X_1, \ldots, X_s$, and formulas $\varphi_1, \ldots, \varphi_s \in \mathrm{FO}^2[<]$ in ENNF, each with at most one free variable, such that

$$\varphi = \beta(x < y, x = y, x > y, \varphi_1, \ldots, \varphi_s) \quad .$$

Proof. We use induction on the structure of φ. The claim is obvious for a formula $Q_a(x)$ with $a \in \Sigma$, for its negation, and for the formulas $x < y$, $x = y$ and $x > y$. For boolean combinations we only need to apply the inductive hypothesis to the parts of the boolean combination. Finally, formulas of the form $\exists x\, \psi$ or $\neg\exists x\, \psi$ have at most one free variable. □

Definition 2.1.7. Let $\varphi \in \mathrm{FO}^2[<]$ in ENNF and let χ be an order formula. Let β and $\varphi_1, \ldots, \varphi_s$ as in Proposition 2.1.6. The *order restriction* $\varphi \restriction_\chi$ is the formula

$$\varphi \restriction_\chi := \begin{cases} \beta(\top, \bot, \bot, \varphi_1, \ldots, \varphi_s) & \text{if } \chi \text{ is } x < y \\ \beta(\bot, \top, \bot, \varphi_1, \ldots, \varphi_s) & \text{if } \chi \text{ is } x = y \\ \beta(\bot, \bot, \top, \varphi_1, \ldots, \varphi_s) & \text{if } \chi \text{ is } x > y \end{cases} \quad .$$

While this definition is purely syntactic, the following proposition states that order restrictions work as intended: An order restriction is the semantic restriction of a formula to a specific ordering of the variables x and y.

Proposition 2.1.8. Let $\varphi \in \mathrm{FO}^2[<]$ in ENNF. Then

$$\varphi \equiv (x < y \wedge \varphi \restriction_{x<y}) \vee (x = y \wedge \varphi \restriction_{x=y}) \vee (x > y \wedge \varphi \restriction_{x>y}) \quad .$$

Proof. Let β and $\varphi_1, \ldots, \varphi_s$ be as in Proposition 2.1.6. Suppose that $(w, i, j) \models \varphi$, thus $(w, i, j) \models \beta(x < y, x = y, x > y, \varphi_1, \ldots, \varphi_s)$. Since the formulas $\varphi_1, \ldots, \varphi_s$ have at most one free variable each, their truth value does not depend on the relative ordering of x and y as interpreted by (w, i, j). Thus, if $x < y$, then $(w, i, j) \models \beta(\top, \bot, \bot, \varphi_1, \ldots, \varphi_s)$ and $(w, i, j) \models \varphi \restriction_{x<y}$. The two other cases $x = y$ and $x > y$ are symmetric.

For the other direction, suppose that

$$(w, i, j) \models (x < y \wedge \varphi \restriction_{x<y}) \vee (x = y \wedge \varphi \restriction_{x=y}) \vee (x > y \wedge \varphi \restriction_{x>y}) \quad .$$

If $(w, i, j) \models x < y \wedge \varphi \restriction_{x<y}$, then $i < j$ and $(w, i, j) \models \beta(\top, \bot, \bot, \varphi_1, \ldots, \varphi_s)$. As above, since the truth values of $\varphi_1, \ldots, \varphi_s$ do not depend on the relative ordering of x and y, we have $(w, i, j) \models \beta(x < y, x = y, x > y, \varphi_1, \ldots, \varphi_s)$ and thus $(w, i, j) \models \varphi$. The two other cases are symmetric. $\square$

For the following lemma, we set $\max\{\} := -\infty$ and $\min\{\} := \infty$.

Lemma 2.1.9. Let $\zeta_1(y), \ldots, \zeta_t(y) \in \mathrm{FO}^2[<]$ over the alphabet Σ with y as the only free variable and in ENNF, and let $w \in \Sigma^+$ be a finite word. Let β be a positive boolean formula in the variables $Z_<, Z_=, Z_>, Y_1, \ldots, Y_t$, let

$$\psi(x, y) = \beta(x < y, x = y, x > y, \zeta_1(y), \ldots, \zeta_t(y)) \quad ,$$

and let $\varphi(x) = \exists y\, \psi(x, y)$. Let

$$p := \max\{j \in |w| \mid (w, j/y) \models \psi(x,y) \upharpoonright_{x<y}\}$$
$$q := \min\{j \in |w| \mid (w, j/y) \models \psi(x,y) \upharpoonright_{x>y}\} \quad .$$

Then for all $i \in |w|$, $(w,i) \models \varphi(x)$ iff $i < p$ or $i > q$ or $(w, i/y) \models \psi(x,y) \upharpoonright_{x=y}$.

Proof. If $(w,i) \models \varphi(x)$, then there is a $j \in |w|$ such that $(w,i,j) \models \psi(x,y)$. If $i < j$, then $(w, j/y) \models \psi(x,y) \upharpoonright_{x<y}$ and $j \leq p$, thus $i < p$. Similarly, if $i > j$, then $(w, j/y) \models \psi(x,y) \upharpoonright_{x>y}$ and $j \geq q$, thus $i > q$. Finally, if $i = j$, then $(w, j/y) = (w, i/y)$ and $(w, i/y) \models \psi(x,y) \upharpoonright_{x=y}$.

Arguing in the other direction, if $i < p$, then $(w,i,p) \models \psi(x,y)$ since $(w, p/y) \models \psi(x,y) \upharpoonright_{x<y}$, and thus $(w,i) \models \varphi(x)$. Similarly, if $i > q$, then $(w,i,q) \models \psi(x,y)$ since $(w, q/y) \models \psi(x,y) \upharpoonright_{x>y}$, and thus $(w,i) \models \varphi(x)$. If $(w, i/y) \models \psi(x,y) \upharpoonright_{x=y}$, then $(w, i, i/y) \models \psi(x,y)$ and thus $(w,i) \models \varphi(x)$. $\square$

Lemma 2.1.10. Let $\varphi \in \mathrm{FO}^2[<]$ over the alphabet Σ in ENNF and with one free variable, let $w \in \Sigma^\star$ be a word, and let $a \in \Sigma$. There is a set $S \subseteq |w|$ which is the union of at most $|\varphi|^2$ intervals such that for every $i \in |w|^a$, $(w,i) \models \varphi(x)$ iff $i \in S$.

Proof. We use induction on the structure of φ. If $\varphi(x)$ is of the form $Q_b(x)$ for some $b \in \Sigma$, then we choose $S = |w|$ if $b = a$ and $S = \emptyset$ otherwise. If $\varphi(x) = \psi_1(x) \vee \psi_2(x)$, then let S_1 and S_2 be the sets for $\psi_1(x)$ and $\psi_2(x)$, respectively, using the inductive hypothesis. We choose $S = S_1 \cup S_2$, which can be constructed as the union of at most as many intervals as were used for the sets S_1 and S_2. Similarly, if $\varphi(x) = \psi_1(x) \wedge \psi_2(x)$, we choose $S = S_1 \cap S_2$, which as before does not introduce any new interval boundaries. If $\varphi(x) = \neg\psi_1(x)$, we complement the set S_1, which adds at most one additional interval.

The interesting case is where $\varphi(x) = \exists y\, \psi(x,y)$. Using Proposition 2.1.6, there is a positive boolean formula β in the variables $Z_<, Z_=, Z_>, X_1, \ldots, X_s, Y_1, \ldots, Y_t$ that uses each of the variables $X_1, \ldots, X_s$ at most once, and there are formulas in x $\xi_1, \ldots, \xi_s$, and formulas in y $\zeta_1, \ldots, \zeta_t$, such that

$$\psi(x,y) = \beta(x < y, x = y, x > y, \xi_1(x), \ldots, \xi_s(x), \zeta_1(y), \ldots, \zeta_t(y)) \quad .$$

Applying the inductive hypothesis to the formulas ξ_σ, $\sigma \in [1,s]$, let S_σ be the set as described in the statement of this lemma, and let $I_{(\sigma,1)}, \ldots, I_{(\sigma,k_\sigma)}$ be intervals such that $k_\sigma \leq |\xi_\sigma|^2$ and $S_\sigma = \bigcup_{\ell=1}^{k_\sigma} I_{(\sigma,\ell)}$. To make it easier to deal with the left and right boundaries of all these intervals, without loss of generality all intervals will be of the form $[f,g)$, i.e. including the left boundary f, but excluding the right boundary g. Let F_σ be the set of all left interval boundaries from the intervals $I_{(\sigma,1)}, \ldots, I_{(\sigma,k_\sigma)}$, and let G_σ be the set of all right boundaries from these intervals. We also define sets F and G of all left and right interval boundaries, respectively: $F := \bigcup_{\sigma\in[1,s]} F_\sigma$ and $G := \bigcup_{\sigma\in[1,s]} G_\sigma$. Because the first and last position of w are not necessarily part of $F \cup G$ already but might be necessary as boundaries for the following construction, we define the set $H := F \cup G \cup \{1, \|w\|+1\}$.

Looking at each interval I defined by two consecutive positions from H, the truth values of the formulas $\xi_1, \ldots, \xi_s$ remain constant among all points from I^a. Let $\xi_1^\star, \ldots, \xi_s^\star$ be these respective truth values. Thus, on all positions from I^a, $\varphi(x)$ is equivalent to $\exists y\, \beta(x < y, x = y, x > y, \xi_1^\star, \ldots, \xi_s^\star, \zeta_1(y), \ldots, \zeta_t(y))$. This formula satisfies the requirements of Lemma 2.1.9, so that with p and q as in the lemma, the truth of $\varphi(x)$ over I^a is determined by the relative position of x with respect to p and q and by the truth of the formulas $\zeta_1(x), \ldots, \zeta_t(x)$ for the positions in between p and q. Putting this all together, we can construct the set S of all positions from $|w|^a$ where $\varphi(x)$ is true as the union of intervals built from *new* interval endpoints, where a new interval endpoint is either an old interval endpoint from H, an interval endpoint that results from this lemma applied to the formulas $\zeta_1(x), \ldots, \zeta_t(x)$, or a point p (as a new right interval boundary) or $q+1$ (as a new left interval boundary) from the application of Lemma 2.1.9 for each interval I. Thus the number of new interval boundaries is at most

$$3|H| + 2 \sum_{\tau \in [1,t]} |\zeta_t|^2 \leq 3 \left(2 \sum_{\sigma \in [1,s]} |\xi_\sigma|^2 + 2 \right) + 2 \sum_{\tau \in [1,t]} |\zeta_\tau|^2 \quad ,$$

which means that S is the union of at most $3 + 3 \sum_{\sigma \in [1,s]} |\xi_\sigma|^2 + \sum_{\tau \in [1,t]} |\zeta_t|^2$ intervals. Unfortunately, this is not necessarily bounded above by $|\varphi|^2$, e.g. when $|\xi_1| \geq \frac{1}{\sqrt{3}}|\varphi|$.

We now develop a better upper bound on the number of new interval boundaries excluding the boundaries from the formulas $\zeta_1, \ldots, \zeta_t$. Let C be the set of all new interval boundaries that are positions from H or points p or $q+1$ from Lemma 2.1.9. We account for these points p and $q+1$ at the interval boundaries of their respective enclosing intervals: Considering each interval $I = [c, d)$ defined by two consecutive interval boundaries c and d from H, we assign the cost of a point p (if it lies inside this interval) to the left interval boundary c, and the cost of point $q+1$ (if it lies inside this interval) to the right interval boundary d. From now on, we refer to these two points as $p(c)$ and $q(d)$, respectively. ($q(d)$ is the point $q+1$ to the left of d.) Thus any point $i \in H$ can contribute at most three new interval boundaries: the point itself, a point $p(i)$ if $p(i)$ is inside the interval that starts at i, and a point $q(i)$ if $q(i)$ is inside the interval that ends at i. Let $C(i)$ be the set of all new interval boundaries contributed by i, and define

$$P(i) := \{p(i) \mid p(i) \in C \text{ and } p(i) < \min\{j \in H \mid j > i\}\}$$
$$Q(i) := \{q(i) \mid q(i) \in C \text{ and } q(i) < \max\{j \in H \mid j < i\}\} \quad .$$

Thus $P(i)$ contains the point $p(i)$ only if it lies inside of the relevant interval and only if it is used as a new interval boundary, and we have $C(i) \subseteq \{i\} \cup P(i) \cup Q(i)$. With

$$\hat{C}(i) := \begin{cases} P(i) \cup Q(i) & \text{if } P(i) \neq \emptyset \text{ and } Q(i) \neq \emptyset \\ \{i\} \cup P(i) \cup Q(i) & \text{otherwise} \end{cases}$$

we have $C(i) \subseteq \hat{C}(i)$, since if both $P(i) \neq \emptyset$ and $Q(i) \neq \emptyset$, then $q(i)$ is within the interval that ends at i and $p(i)$ is in the interval that starts at i. Thus $\varphi(x)$ is true on all points from $[q(i), p(i))^a$, and i itself is not a new interval boundary.

We now split up the contributions of new interval boundaries C to left and right interval boundaries F and G. For every $\sigma \in [1, s]$, and all $i \in F_\sigma$, we define the set

$$\hat{C}_F(\sigma, i) := \begin{cases} \{i\} \cup P(i) & \text{if } i \in G \\ C(i) & \text{otherwise} \end{cases} \quad .$$

Similarly, for all $i \in G_\sigma$, we define the set

$$\hat{C}_G(\sigma, i) := \begin{cases} \{i\} \cup Q(i) & \text{if } i \in F \\ C(i) & \text{otherwise} \end{cases} \quad .$$

We claim that

$$C \subseteq \{1, \|w\|+1\} \cup P(1) \cup Q(\|w\|+1) \cup \bigcup_{\sigma \in [1,s]} \left(\bigcup_{i \in F_\sigma} \hat{C}_F(\sigma, i) \cup \bigcup_{i \in G_\sigma} \hat{C}_G(\sigma, i) \right) \quad .$$

To see this, we argue that for every $i \in H$, $C(i)$ is contained in the set on the right. For $i = 1$, we have $C(i) \subseteq \{i\} \cup P(i)$ and for $i = \|w\|+1$, we have $C(i) \subseteq \{\|w\|+1\} \cup Q(\|w\|+1)$. For $i \in F - G$ and $i \in G - F$ the inclusion holds by the definition of $\hat{C}_F(\sigma, i)$ and $\hat{C}_G(\tau, i)$. Finally, for $i \in F \cap G$ we have $\hat{C}_F(\sigma, i) \cup \hat{C}_G(\tau, i) = \{i\} \cup P(i) \cup Q(i)$.

To bound the size of the sets $\hat{C}_F(\sigma, i)$ and $\hat{C}_G(\tau, i)$, we define the sets of interval boundaries from all other subformulas in x: $F_{\overline{\sigma}} = \bigcup_{\tau \in [1,s,] \tau \neq \sigma} F_\tau$, and $G_{\overline{\sigma}} = \bigcup_{\tau \in [1,s], \tau \neq \sigma} G_\tau$. Fixing any two consecutive left interval boundaries c and d from $F_{\overline{\sigma}}$, we make the following two observations about the contributions $\hat{C}_F(\sigma, i)$ of the points $i \in F_\sigma \cap [c, d)$.

- There is at most one $i \in F_\sigma \cap [c, d)$ with $P(i) \neq \emptyset$. To see this, suppose that there are $i \in F_\sigma \cap [c, d)$ with $P(i) \neq \emptyset$ and $j \in F_\sigma \cap [c, d)$ with $j < i$. Since the interval $[c, d)$ contains no left interval boundaries besides the ones from F_σ, all formulas from $\xi_1(x), \ldots, \xi_s(x)$ that are true over the interval starting at i are also true over the interval starting at j. Since β is a positive in $X_1, \ldots, X_s$, for all ℓ from the interval that starts with j with $w_\ell = a$ we have $(w, \ell, p(i)) \models \psi(x, y)$, and thus $P(j) = \emptyset$.

- For any $i \in F_\sigma \cap [c, d)$ with $Q(i) \neq \emptyset$ and $Q(i) \subseteq \hat{C}_F(\sigma, i)$, $i \notin \hat{C}_F(\sigma, i)$. To see this, let $i \in F_\sigma \cap [c, d)$ with $Q(i) \neq \emptyset$ and $Q(i) \subseteq \hat{C}_F(\sigma, i)$. Thus, by the definition of $\hat{C}_F(\sigma, i)$, $i \notin G$, i.e. i is not a right interval boundary. Therefore all formulas from $\xi_1(x), \ldots, \xi_s(x)$ that are true over the interval that ends at i are also true over the interval starting at i. Thus, since β is positive in $X_1, \ldots, X_s$, for all ℓ from the interval that starts with i with $w_\ell = a$ we have $(w, \ell, q(i)) \models \psi(x, y)$. Thus i is not a new interval boundary, and since $\hat{C}_F(\sigma, i) = C(i)$, $i \notin \hat{C}_F(\sigma, i)$.

Thus among all points $i \in F_\sigma \cap [c, d)$, $\hat{C}_F(\sigma, i)$ contains at most one element, except for at most one i where $\hat{C}_F(\sigma, i)$ contains at most two elements. Therefore the set $\bigcup_{i \in F_\sigma \cap [c,d)} \hat{C}_F(\sigma, i)$ has at most $|F_\sigma \cap [c, d)| + 1$ elements. Thus for any $\sigma \in [1, s]$, we have

$$\sum_{i \in F_\sigma} |\hat{C}_F(\sigma, i)| \leq \sum_{c \in F_{\overline{\sigma}}} (|F_\sigma \cap [c, d)| + 1) = |F_{\overline{\sigma}}| + \sum_{c \in F_{\overline{\sigma}}} |F_\sigma \cap [c, d)| = |F_{\overline{\sigma}}| + |F_\sigma| \quad .$$

Combining this with the observation that any set $\hat{C}_F(\sigma, i)$ contains at most two elements, we have

$$
\begin{aligned}
\sum_{i\in F_\sigma} |\hat{C}_F(\sigma,i)| &\le \min\left\{2|F_\sigma|, |F_{\overline{\sigma}}| + |F_\sigma|\right\} \\
&= |F_\sigma| + \min\left\{|F_\sigma|, |F_{\overline{\sigma}}|\right\} \\
&\le |\xi_\sigma|^2 + \min\left\{|\xi_\sigma|^2, \sum_{\tau\in[1,s,]\tau\neq\sigma} |\xi_\tau|^2\right\} \\
&\le |\xi_\sigma|^2 + \min\left\{|\xi_\sigma|^2, \left(\sum_{\tau\in[1,s],\tau\neq\sigma} |\xi_\tau|\right)^2\right\} \\
&\le |\xi_\sigma|^2 + |\xi_\sigma| \sum_{\tau\in[1,s],\tau\neq\sigma} |\xi_\tau| \qquad = |\xi_\sigma| \sum_{\tau\in[1,s]} |\xi_\tau| \quad .
\end{aligned}
$$

By a symmetric argument, we have

$$
\sum_{i\in G_\sigma} |\hat{C}_G(\sigma,i)| = |\xi_\sigma| \sum_{\tau\in[1,s]} |\xi_\tau| \quad .
$$

Thus

$$
\begin{aligned}
|C| &\le 4 + \left|\bigcup_{\sigma\in[1,s]} \left(\bigcup_{i\in F_\sigma} \hat{C}_F(\sigma,i) \cup \bigcup_{i\in G_\sigma} \hat{C}_G(\sigma,i)\right)\right| \\
&\le 4 + \sum_{\sigma\in[1,s]} \left(\sum_{i\in F_\sigma} |\hat{C}_F(\sigma,i)| + \sum_{i\in G_\sigma} |\hat{C}_G(\sigma,i)|\right) \\
&\le 4 + \sum_{\sigma\in[1,s]} \left(2|\xi_\sigma| \sum_{\tau\in[1,s]} |\xi_\tau|\right) \qquad = 4 + 2\left(\sum_{\sigma\in[1,s]} |\xi_\sigma|\right)^2 \quad .
\end{aligned}
$$

Therefore the total number of new interval boundaries is at most

$$
\begin{aligned}
|C| + \sum_{\tau\in[1,t]} 2|\zeta_\tau|^2 &\le 4 + 2\left(\sum_{\sigma\in[1,s]} |\xi_\sigma|\right)^2 + 2\sum_{\tau\in[1,t]} |\zeta_\tau|^2 \\
&\le 2\left(2 + \left(\sum_{\sigma\in[1,s]} |\xi_\sigma|\right)^2 + \left(\sum_{\tau\in[1,t]} |\zeta_\tau|\right)^2\right) \\
&\le 2\left(2 + \sum_{\sigma\in[1,s]} |\xi_\sigma| + \sum_{\tau\in[1,t]} |\zeta_\tau|\right)^2 \qquad \le 2\cdot|\varphi|^2 \quad .
\end{aligned}
$$

Thus we have at most $|\varphi|^2$ intervals. $\square$

While we suspect that this quadratic bound on the number of intervals is not optimal, we do have an explicit construction of a formula that requires a super-linear number of intervals.

Proof of Theorem 2.1.1. We assume that φ is in ENNF, otherwise we can easily convert it into ENNF while at most doubling its size. Let w be a model of φ, and let $\varphi_1, \ldots, \varphi_k$ be the subformulas of φ of the form $\exists x\, \psi$ for some variable x and some formula ψ. For every $\kappa \in [1, k]$, we use Proposition 2.1.6 to find a positive boolean formula β in the variables $Z_<, Z_=, Z_>, X_1, \ldots, X_s, Y_1, \ldots, Y_t$, and formulas in x $\xi_1, \ldots, \xi_s$, and formulas in y $\zeta_1, \ldots, \zeta_t$, such that $\varphi_\kappa(x) = \exists y\, \psi_\kappa(x, y)$, where

$$\psi_\kappa(x, y) := \beta(x < y, x = y, x > y, \xi_1(x), \ldots, \xi_s(x), \zeta_1(y), \ldots, \zeta_t(y)) \quad .$$

We will use Lemma 2.1.10 to bound the number of possible combinations of truth values for the formulas $\xi_1(x), \ldots, \xi_s(x)$, and construct a set W_κ of witnesses (interpretations of y) to satisfy the formula $\psi(x, y)$ for every fixed combination of truth values for the formulas $\xi_1(x), \ldots, \xi_s(x)$. We will see that for any of these combinations of truth values, we only need at most two witnesses. Then we will show that the structure w restricted to all witnesses is still a model of φ.

For every $a \in \Sigma$ and every $\sigma \in [1, s]$, let S^a_σ be a set as in Lemma 2.1.10 applied to the formula $\xi_\sigma(x)$ and a, where S^a_σ is the union of at most $|\xi_\sigma|^2$ intervals. Thus there is a set $\mathfrak{I}^a_\kappa$ of at most $2 + \sum_{\sigma \in [1,s]} |\xi_\sigma|^2$ intervals such that for any interval $I \in \mathfrak{I}^a_\kappa$, there are $\xi^I_1, \ldots, \xi^I_s \in \{\top, \bot\}$ such that $\bigcup_{I \in \mathfrak{I}^a_\kappa} = |w|$ and for all $i \in I^a$, we have $(w, i) \models \xi_\sigma(x)$ iff $\xi^I_\sigma = \top$. For each $I \in \mathfrak{I}^a_\kappa$, we apply Lemma 2.1.9 to the formula $\varphi^I_\kappa(x) = \exists y\, \psi^I_\kappa(x, y)$, where

$$\psi^I_\kappa(x, y) := \beta(x < y, x = y, x > y, \xi^I_1, \ldots, \xi^I_s, \zeta_1(y), \ldots, \zeta_t(y)) \quad ,$$

and let p_I, q_I be as in that lemma. Let $W^a_\kappa := \bigcup_{I \in \mathfrak{I}^a_\kappa} \{p_I, q_I\}$, let $W := \bigcup_{\kappa \in [1,k], a \in \Sigma} W^a_\kappa$, and let w_W be the restriction of w to the positions from W such that $|w| = W$. We claim that $w_W \models \varphi$. To see this, we show that for every subformula η of φ and all $i, j \in W$, $(w, i, j) \models \eta$ iff $(w_W, i, j) \models \eta$.

- If η is atomic formula the claim is obvious.

- If η is a boolean combination of α and β, then the claim follows immediately by applying the inductive hypothesis to α and β.

- If η starts with an existential quantifier, then $\eta = \varphi_\kappa(x)$ for some $\kappa \in [1, k]$, using the enumeration of these subformulas from above.

  Suppose that $i \in W$ and $(w, i) \models \eta(x)$. Then there is a $j \in |w|$ such that $(w, i, j) \models \psi_\kappa(x, y)$. Let $a = w_i$ and let $I \in \mathfrak{I}^a_\kappa$ such that $i \in I$. We find $\hat{j} \in W$ such that $(w, i, \hat{j}) \models \psi(x, y)$ as follows: If $j < i$ then $\hat{j} = q_I$, if $j > i$ then $\hat{j} = p_I$, and if $j = i$ then $\hat{j} = i$. Applying the inductive hypothesis, we see that $(w_W, i, \hat{j}) \models \psi(x, y)$, and thus $(w_W, i) \models \eta(x)$.

  Suppose that $i \in W$ and $(w_W, i) \not\models \eta(x)$. Then there is a $j \in W$ such that $(w_W, i, j) \models \psi_\kappa(x, y)$. Using the inductive hypothesis, we have $(w, i, j) \models \psi_\kappa(x, y)$ and thus $(w, i) \models \eta(x)$.

So because $w \models \varphi$, we have $w_W \models \varphi$. The size of W is at most $2 \sum_{\kappa \in [1,k], a \in \Sigma} \mathfrak{I}^a_\kappa$, where $k \leq |\varphi|$ and $|\mathfrak{I}^a_\kappa| \leq 2 + |\varphi_\kappa|^2 \leq 2 + |\varphi|^2$. Thus $|W| = O(|\Sigma| \cdot |\varphi|^3)$. □

2.2 Satisfiability of FO^2 on Finite Words and Finite Power Words

In this section we first present known results on the complexity of the satisfiability problem for FO^2[] on monadic structures, and FO^2[<] and FO^2[<, Suc] on both finite words and finite power words. We conclude with our main result, an NP algorithm for satisfiability of $\mathrm{FO}^2[<]$ on finite words.

A proof sketch of Theorem 2.2.1 is presented in [10], building on [11,26], and using a reduction from a tiling problem. We present an alternative, although essentially equivalent, full proof that directly encodes the computation of a nondeterministic Turing Machine. We also present an explicit and full proof for Theorem 2.2.3, which has been stated in [10]. Theorem 2.2.6, which also appears here with a more detailed proof, is from [47].

The language $\mathrm{FO}^2[]$ does not contain any numeric predicates. On monadic structures, all we have is equality and the monadic predicates. Thus our structures are collections of points which satisfy some of the monadic predicates. Since $\mathrm{FO}^2[]$ is contained in both $\mathrm{FO}^2[<]$ and $\mathrm{FO}^2[<, \mathrm{Suc}]$, the following theorem implies NEXP-hardness for these more expressive languages.

Theorem 2.2.1. [10, 11, 26] Satisfiability for $\mathrm{FO}^2[]$ on monadic structures is NEXP-hard.

Proof. Let $M = (Q, \Sigma, \delta, q_0, F)$ be a nondeterministic Turing Machine that runs in time 2^{n^k} for some $k \in \mathbb{N}$ on all inputs of length n. Here Q is the set of states, Σ is the input alphabet, $\Gamma := \Sigma \mathbin{\dot\cup} \{\triangleright, \sqcup\}$ is the tape alphabet, $\delta : Q \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{-1, 0, 1\})$ is the nondeterministic state transition function, q_0 is the initial state, and F is a set of accepting states. We describe a function f that maps inputs for M to FO^2 formulas such that for every input $w \in \Sigma^\star$, $f(w)$ is satisfiable iff w is accepted by M.

An element of the universe of any model satisfying the formula $f(w)$ represents a single tape cell at a single time step in the computation of M. The formula $f(w)$ uses the following unary predicates: $T_1, \ldots, T_{n^k}$ for a binary encoding of each time step; $P_1, \ldots, P_{n^k}$ for a binary encoding of each tape cell; $C_a, a \in \Gamma$ for the symbols in the tape alphabet; $S_q, q \in Q$ for the states of M.

To encode the computation of M, we need to be able to say "at the next time step", "on the previous tape cell" and "on the next tape cell". Because we do not have numeric predicates, we use the following boolean formula to say that the number encoded by the boolean variables $X_1, \ldots, X_k$ in binary is one smaller than the number encoded by $Y_1, \ldots, Y_k$.

$$\begin{aligned}&\mathrm{NEXT}\left(X_1, \ldots, X_k, Y_1, \ldots, Y_k\right) \\ &\qquad = \bigvee_{i \in [1,k]} \left(\bigwedge_{j \in [1,i)} (X_j \leftrightarrow Y_j) \wedge \neg X_i \wedge Y_i \wedge \bigwedge_{j \in (i,k]} (X_j \wedge \neg Y_j) \right)\end{aligned}$$

We also use a boolean formula to check for equality of two sequences of boolean variables.

$$\mathrm{EQUALS}\left(X_1, \ldots, X_k, Y_1, \ldots, Y_k\right) = \bigwedge_{i \in [1,k]} X_i \leftrightarrow Y_i$$

We construct the formula $f(w)$ as the conjunction of all the following formulas. We use $\overline{T}(x)$ as a shorthand for $T_1(x), \ldots, T_{2^{n^k}}(x)$, and also use $\overline{P}(x)$, $\overline{\bot}$ and $\overline{\top}$ in a similar way. The binary representation of a number $i \in \mathbb{N}$ as a sequence of $\top$ and $\bot$ is denoted as $\langle i \rangle$.

- The universe contains elements for every part of the computation, i.e. there is an element for time step 0 and tape position 0, every tape cell has a successor in time except at the end of the computation, and every tape cell has another tape cell to its right, except if it is the right-most tape cell.

$$\exists x \left(\mathrm{EQUALS}\left(\overline{T}(x), \langle 0 \rangle\right) \wedge \mathrm{EQUALS}\left(\overline{P}(x), \langle 0 \rangle\right)\right)$$

$$\forall x. \neg \mathrm{EQUALS}\left(\overline{T}(x), \overline{1}\right) \ \exists y \left(\mathrm{NEXT}\left(\overline{T}(x), \overline{T}(y)\right) \wedge \mathrm{EQUALS}\left(\overline{P}(x), \overline{P}(y)\right)\right)$$

$$\forall x. \neg \mathrm{EQUALS}\left(\overline{P}(x), \overline{1}\right) \ \exists y \left(\mathrm{EQUALS}\left(\overline{T}(x), \overline{T}(y)\right) \wedge \mathrm{NEXT}\left(\overline{P}(x), \overline{P}(y)\right)\right)$$

- Every tape cell contains at most one tape symbol.

$$\forall x \left(\bigwedge_{a \neq b \in \Gamma} \neg (C_a(x) \wedge C_b(x)) \right)$$

- At every time step, at most one tape cell has a state marker.

$$\forall x \forall y. \left(\text{EQUALS}\left(\overline{T}(x), \overline{T}(y)\right) \wedge \neg \text{EQUALS}\left(\overline{P}(x), \overline{P}(y)\right)\right)$$
$$\bigwedge_{q,p \in Q} \neg (S_q(x) \wedge S_p(y))$$

- At the initial time step, the tape contains the input followed by blank symbols, the head points to the first symbol of the input, and M is in the initial state.

$$\begin{aligned}\forall x.\text{EQUALS}\left(\overline{T}(x), \langle 0 \rangle\right) \Bigg(& \left(\text{EQUALS}\left(\overline{P}(x), \langle 0 \rangle\right) \rightarrow C_{\triangleright}(x)\right) \\ & \wedge \bigwedge_{i \in [1,n]} \left(\text{EQUALS}\left(\overline{P}(x), \langle i \rangle\right) \rightarrow C_{w_i}(x)\right) \\ & \wedge \left(\left(\bigwedge_{i \in [0,n)} \neg \text{EQUALS}\left(\overline{P}(x), \langle i \rangle\right) \right) \rightarrow C_{\sqcup}(x) \right) \\ & \wedge \left(\text{EQUALS}\left(\overline{P}(x), \langle 1 \rangle\right) \rightarrow S_{q_0}(x)\right) \Bigg)\end{aligned}$$

- The contents of any tape cell not at the current head position remains unchanged.

$$\forall x. \left(\bigwedge_{q \in Q} \neg S_q(x) \right) \forall y. \left(\text{NEXT}\left(\overline{T}(x), \overline{T}(y)\right) \wedge \text{EQUALS}\left(\overline{P}(y), \overline{P}(x)\right)\right)$$
$$\bigwedge_{a \in \Gamma} (C_a(x) \leftrightarrow C_a(y))$$

- The tape contents at the head position together with the current state determine the tape contents of the same cell and the head position at the next time step.

$$\forall x \bigwedge_{a\in\Gamma, q\in Q} \Bigg((C_a(x) \wedge S_q(x)) \rightarrow \forall y. \left(\mathrm{NEXT}\left(\overline{T}(x), \overline{T}(y)\right) \wedge \mathrm{EQUALS}\left(\overline{P}(y), \overline{P}(x)\right)\right) \bigvee_{(a',q',d)\in\delta(a,q)} (C_{a'}(y) \wedge \varphi_{d,q'}(y)) \Bigg)$$

where

$$\varphi_{d,q'}(y) := \begin{cases} \forall x. \left(\tau(x,y) \wedge \mathrm{NEXT}\left(\overline{P}(x), \overline{P}(y)\right)\right) \; S_{q'}(x) & \text{if } d = -1 \\ S_{q'}(y) & \text{if } d = 0 \\ \forall x. \left(\tau(x,y) \wedge \mathrm{NEXT}\left(\overline{P}(y), \overline{P}(x)\right)\right) \; S_{q'}(x) & \text{if } d = 1 \end{cases}$$

and $\tau(x, y) := \mathrm{EQUALS}\left(\overline{T}(x), \overline{T}(y)\right)$

- The final state of the computation is accepting.

$$\exists x \left(\mathrm{EQUALS}\left(\overline{T}(x), \overline{1}\right) \wedge \bigvee_{q\in F} S_q(x) \right)$$

We observe that if w is accepted by M, then the corresponding computation can be converted into a model of the formula $f(w)$. If $f(w)$ has a model, then this model encodes an accepting computation of M, and thus w is accepted by M. □

Corollary 2.2.2. Satisfiability for $\mathrm{FO}^2[<]$ and $\mathrm{FO}^2[<, \mathrm{Suc}]$ on finite power words is NEXP-hard.

A similar corollary for finite words does not follow directly from Theorem 2.2.1, because for finite words we have the restrictions that no two unary predicates can hold at the same position. However, in the presence of a successor relation, we can easily work around this semantic restriction.

Theorem 2.2.3. [10] Satisfiability for $\mathrm{FO}^2[\mathrm{Suc}]$ on structures with only one unary predicate is NEXP-hard.

Proof. We present a reduction to the satisfiability problem for $\mathrm{FO}^2[]$ on monadic structures. NEXP-hardness then follows from Theorem 2.2.1.

We convert a given FO^2 formula φ that uses only the unary predicates $X_1, \dots, X_k$ into a $\mathrm{FO}^2[\mathrm{Suc}]$ formula $f(\varphi)$ using only one unary predicate X such that φ is satisfiable iff $f(\varphi)$ is satisfiable. The main idea is to distribute the information about which ones of the k predicates hold for a given element of the universe onto a sequence of $2k+2$ new elements, where, as a marker, X does not hold for the first two elements of the sequence, X holds at all other odd positions, and the remaining even positions indicate whether X_i holds or not. For example, if $k = 3$ and for some structure A we have $a \in X_1^A$, $a \in X_2^A$ and $a \notin X_3^A$, we translate this into a structure B with only one unary relation X^B and elements $a_1, \dots, a_8$, occurring in this order with respect to the successor relation. We put only $a_3, \dots, a_7$ into X^B, so that a binary representation of this sequence is 00111110 when writing 0 for the positions where X does not hold and 1 for the positions where it holds.

We will describe this translation in more detail after defining the following two helper formulas. $\varphi_{\mathrm{mark}}(x)$ says that the string 001 starts at x, marking the beginning of an encoding sequence corresponding to one element from the original structure. The formula $\varphi_i(x)$ says that $2i+1$ steps to the left of x, X holds. Assuming that x is the beginning of an encoding sequence, this is the position that encodes the truth value of $X_i(x)$.

$$\varphi_{\mathrm{mark}} := \neg X(x) \wedge \exists y\,(\mathrm{Suc}(x,y) \wedge \neg X(y)) \wedge \exists x\,(\mathrm{Suc}(y,x) \wedge X(x))$$
$$\varphi_i := \exists y.\mathrm{Suc}(x,y)\ [\exists x.\mathrm{Suc}(y,x)\,\exists y.\mathrm{Suc}(x,y)]^i\ X(y)$$

We recursively construct the formula $f(\varphi)$ as follows.

- If $\varphi = X_i(x)$, then $f(\varphi) = \varphi_i(x)$.
- If φ is the formula $x = y$, then $f(\varphi) = \varphi$.
- If $\varphi = \psi \vee \xi$, then $f(\varphi) = f(\psi) \vee f(\xi)$.
- If $\varphi = \neg\psi$, then $f(\varphi) = \neg f(\psi)$.
- If $\varphi = \exists x\, \psi$, then $f(\varphi) = \exists x\, (\varphi_{\mathrm{mark}}(x) \wedge f(\psi))$.

We claim that φ is satisfiable iff $f(\varphi)$ is satisfiable.

If φ is satisfiable, then let A be a model of φ. We construct a model B of $f(\varphi)$. For every element $a \in |A|$, the universe of B contains the elements $a_1, \ldots, a_{2k+2}$. $|B|$ contains no other elements. For all $a \in |A|$, we set $a_1, a_2 \notin X^B$, and for all $i \in [1, k]$, $a_{2i+1} \notin X^B$ and $a_{2i+2} \in X^B$ iff $a \in X_i^A$. We require that for all $a \in |A|$, the sequence $a_1, \ldots, a_{2k+2}$ is consistent with the successor relation of B, and choose an arbitrary ordering of the elements of $|A|$ and thus for the ordering of these sequences.

By induction on the structure of φ, we prove that for all $a, b \in |A|$, if $(A, a, b) \models \varphi$ then $(B, a_1, b_1) \models f(\varphi)$.

- If $\varphi = X_i(x)$, then $a \in X_i^A$. By the definition of X^B, we thus have $a_{2i+2} \in X^B$, hence $(B, a_1) \models \varphi_i(x)$ and thus $(B, a_1, b_1) \models f(\varphi)$.
- If φ is the formula $x = y$, then $a = b$, thus $a_1 = b_1$ and $(B, a_1, b_1) \models f(\varphi)$.
- If $\varphi = \neg\psi$ or $\varphi = \psi \vee \xi$, then we only need to apply the inductive hypothesis to ψ and ξ.
- If $\varphi = \exists x\, \psi$, then there is a $c \in |A|$ such that $(A, c, b) \models \psi$. Using the inductive hypothesis, we thus have $(B, c_1, b_1) \models f(\psi)$. Since by definition $c_1 \notin X^B$, $c_2 \notin X^B$, and $c_3 \in X^B$, we also have $(B, c_1, b_1) \models \varphi_{\mathrm{mark}}(x)$, and thus $(B, a_1, b_1) \models f(\varphi)$.

Thus $B \models f(\varphi)$, and $f(\varphi)$ is satisfiable.

If $f(\varphi)$ is satisfiable, then let B be a model of $f(\varphi)$. We construct a model A of φ. The universe of A is $|A| := \{a \in B \mid (B, a) \models \varphi_{\text{mark}}(x)\}$, and the relations are defined as $X_i^A = \{a \in |A| \mid (B, a) \models \varphi_i(x)\}$.

By induction on the structure of φ, we prove that for all $a, b \in |A|$, if $(B, a, b) \models f(\varphi)$, then $(A, a, b) \models \varphi$.

- If $\varphi = X_i(x)$, then $(B, a, b) \models \varphi_i(x)$, thus $a \in X_i^A$ and $(A, a, b) \models \varphi$.
- If φ is the formula $x = y$, then $a_1 = b_1$, thus $a = b$ and $(A, a, b) \models \varphi$.
- If $\varphi = \neg\psi$ or $\varphi = \psi \vee \xi$, then it suffices to apply the inductive hypothesis to ψ and ξ.
- If $\varphi = \exists x\, \psi$, then $(B, a, b) \models \exists x\, (\varphi_{\text{mark}}(x) \wedge f(\psi))$. Thus there is a $c \in |B|$ such that $(B, c, b) \models \varphi_{\text{mark}}(x)$ and $(B, c, b) \models f(\psi)$. Because $(B, c) \models \varphi_{\text{mark}}(x)$, we have $c \in |A|$, thus the inductive hypothesis applies to ψ, hence $(A, c, b) \models f(\psi)$, and $(A, a, b) \models \varphi$.

Thus $A \models \varphi$, and φ is satisfiable. $\square$

Corollary 2.2.4. Satisfiability for $\mathrm{FO}^2[<, \mathrm{Suc}]$ on finite words is NEXP-hard.

For the proof of the following theorem we refer the reader to [10], where this result is stated for infinite words. The proof is based on a small model property similar to our result from the previous section, but allowing for models of size that is exponential in the quantifier depth of the given formula.

Theorem 2.2.5. [10] Satisfiability for $\mathrm{FO}^2[<, \mathrm{Suc}]$ (and thus for $\mathrm{FO}^2[<]$) on finite power words is in NEXP.

Theorem 2.2.6. Satisfiability for $\mathrm{FO}^2[<]$ on finite words is NP-hard.

Proof. We present a reduction from SAT. Let α be a boolean formula over the variables $X_1, \ldots, X_n$. We construct a $\mathrm{FO}^2[<]$ formula $f(\alpha)$ over a binary alphabet $\{0, 1\}$ such that α is satisfiable if $f(\alpha)$ has a finite word model.

We need the following helper formulas to identify the positions in a finite word of length n: $\mathrm{DIST}_{\min,i}(x)$ to say that the distance between the first position in the word and position x is i, $\mathrm{DIST}_{\max,i}(x)$ to say that the distance between x and the last position in the word is i, and $\mathrm{POS}_i(x)$ to say that x points to position i.

$$
\begin{aligned}
\text{DIST-GE}_{\min,i}(x) &:= \begin{cases} [\exists y.(y < x)\, \exists x.(x < y)]^{(i-1)/2}\ (x = \min) & \text{if } i \text{ odd} \\ [\exists y.(y < x)\, \exists x.(x < y)]^{(i-2)/2}\ \exists y.(y < x)\,(y = \min) & \text{if } i \text{ even} \end{cases} \\
\mathrm{DIST}_{\min,i}(x) &:= \text{DIST-GE}_{\min,i}(x) \wedge \neg \text{DIST-GE}_{\min,i+1}(x) \\
\text{DIST-GE}_{\max,i}(x) &:= \begin{cases} [\exists y.(y > x)\, \exists x.(x > y)]^{(i-1)/2}\ (x = \max) & \text{if } i \text{ odd} \\ [\exists y.(y > x)\, \exists x.(x > y)]^{(i-2)/2}\ \exists y.(y > x)\,(y = \max) & \text{if } i \text{ even} \end{cases} \\
\mathrm{DIST}_{\max,i}(x) &:= \text{DIST-GE}_{\max,i}(x) \wedge \neg \text{DIST-GE}_{\max,i+1}(x) \\
\mathrm{POS}_i(x) &:= \mathrm{DIST}_{\min,i-1}(x) \wedge \mathrm{DIST}_{\max,n-i-1}(x)
\end{aligned}
$$

Now we inductively define the formula $f(\alpha)$.

- If $\alpha = X_i$, then $f(\alpha) = \exists x\, (\mathrm{POS}_i(x) \wedge Q_1(x))$.
- If $\alpha = \beta \vee \gamma$, then $f(\alpha) = f(\beta) \vee f(\gamma)$, similarly for $\alpha = \beta \wedge \gamma$ and $\alpha = \neg\beta$.

If α is satisfiable, then let $a : \{X_1, \ldots, X_n\} \to \{0, 1\}$ be a satisfying assignment to the boolean variables of α. We observe that the finite word $a(X_1) \ldots a(X_n)$ satisfies $f(\alpha)$.

If $f(\alpha)$ is satisfiable, then let w be a finite word such that $w \models f(\alpha)$. If the length of w is different from n, then we claim that the assignment $a : X_1 \mapsto 0, \ldots, X_n \mapsto 0$ satisfies α since $POS_i(x)$ does not for any i and any position of w. Otherwise the assignment a with $a(X_i) = w_i$ satisfies α. $\square$

Theorem 2.2.7. Satisfiability for $\mathrm{FO}^2[<]$ on finite words is in NP.

Proof. Theorem 2.1.1 implies that for any $\varphi \in \mathrm{FO}^2[<]$ over an alphabet Σ, it suffices to guess a model of size $O(|\Sigma| \cdot |\varphi|^3)$ and verify that this model satisfies φ. $\square$

To summarize, we now have a complete picture of the complexity of the satisfiability problem for FO^2 on finite words and finite power words, as illustrated in Table 2.1

	$\mathrm{FO}^2[<]$	$\mathrm{FO}^2[<, \mathrm{Suc}]$
finite words	NP-complete	NEXP-complete
finite power words	NEXP-complete	NEXP-complete

Table 2.1. Complexity of the Satisfiability Problem for FO^2 on Finite Words and Finite Power Words

CHAPTER 3

SUCCINCTNESS OF FO^k

Succinctness is a measure to compare different logics based on the size of their formulas. Intuitively, we say that a logic L_1 is more succinct than a logic L_2 if the formulas in L_1 are shorter than the equivalent formulas in L_2. If L_1 and L_2 do not have the same expressive power, then of course this comparison only makes sense if we restrict our attention to the properties expressible in both logics. For the following definition, we think of $O(f)$, $\Omega(f)$, etc. as function classes.

Definition 3.0.8. Let F be a class of functions with signature $\mathbb{N} \to \mathbb{N}$, and let L_1 and L_2 be logics on a class of structures C. The *succinctness* of L_1 in L_2 on C is F if

(i) there is a function $f \in F$ such that for every sentence $\varphi \in L_1$ there is a sentence in L_2 of length at most $f(|\varphi|)$ that is equivalent to φ on all structures from C, and

(ii) there is a function $f \in F$ such that there is a sentence $\varphi \in L_1$ that is expressible in L_2 on C-structures such that any L_2 sentence equivalent to φ has length at least $f(|\varphi|)$.

We observe that succinctness is only defined if every property of structures from C expressible in L_1 is also expressible in L_2. To compare logics of different expressive power, we restrict one of the two logics semantically to the properties expressible in the other logic.

Our definition of succinctness encompasses both an upper bound and a lower bound, allowing us to state succinctness results more naturally than with the original definition from [15]. Similar notions of succinctness have been implicitly used before [1, 24, 35, 48]. With our definition, using function classes $O(\cdot)$ and $o(\cdot)$ only invokes the upper bound from condition (i), $\Omega(\cdot)$ and $\omega(\cdot)$ only invokes the lower bound from condition (ii), and $\Theta(\cdot)$ invokes both conditions. It should also be noted that even though for some logics succinctness might be well defined, a function $f : \mathbb{N} \to \mathbb{N}$ such that the succinctness is $\Theta(f)$ does not necessarily exist.

The earliest investigations on succinctness include a result on the non-elementary succinctness of first-order logic in the temporal logic LTL on finite words [24, 35]. More recent work considered the succinctness of the temporal logic CTL$^+$ in CTL. Wilke proved a lower bound of $2^{\Omega(n)}$[48], which was later improved to $\Theta(n!)$ [1]). The succinctness of monadic second-order logic and similar logics on finite trees [14] has also been investigated, inspired by an interest in XML query languages.

After exploring connections between the succinctness of first-order formulas and complexity theory in Section 3.1, we survey known results and techniques for proving bounds on succinctness in Section 3.2, and conclude this chapter with Section 3.3 in which we present a new succinctness bound on a simple first-order property .

3.1 Succinctness and Complexity Theory

Besides a genuine model-theoretic interest in the succinctness of first-order languages, our investigation is driven by our desire to understand the complexity-theoretic trade-off between parallel time and the number of processors. We recall the following descriptive complexity definition of formulas built from iterated quantifier blocks.

Definition 3.1.1. For any function $t : \mathbb{N} \to \mathbb{N}$, $\mathrm{FO}[t(n)]$ is the set of all first-order formulas of the form $[QB]^{t(n)}M_0$, i.e. $t(n)$ identical copies of QB followed by M_0, where $QB = Q_1x_1.M_1 \ldots Q_kx_k.M_k$, $Q_1, \ldots, Q_k$ are quantifiers, and $M_0, \ldots, M_k$ are quantifier-free formulas. Furthermore, $\mathrm{FO}^k[t(n)]$ is the fragment of $\mathrm{FO}[t(n)]$ of formulas using at most k variables, and FO and FO^k are the corresponding sets of formulas for arbitrary t. We define the set of second-order formulas $\mathrm{SO}[t(n)]$ analogously.

Theorem 3.1.2. [20, 22] For all $k > 1$, $\mathrm{FO}^k = \mathrm{DSPACE}[n^{k-1}]$.

This early theorem on the expressive power of iterated quantifier block formulas started a more detailed investigation of the complexity theoretic meaning of these classes, summarized in Figure 3.1.

$$
\begin{array}{ccccc}
\mathrm{FO}[1] & = & \mathsf{AC}^0 & & \\
\cap\!| & & \cap\!| & & \\
\mathrm{FO}[\frac{\log n}{\log\log n}] & \supseteq & \mathsf{NC}^1 & & \\
 & & \cap\!| & & \\
 & & \mathsf{L} & & \\
 & & \cap\!| & & \\
\cap\!| & & \mathsf{NL} & & \\
 & & \cap\!| & & \\
 & & s\mathsf{AC}^1 & & \\
 & & \cap\!| & & \\
\mathrm{FO}[\log n] & = & \mathsf{AC}^1 & & \\
\cap\!| & & \cap\!| & & \\
\mathrm{FO}[n^{O(1)}] & = & \mathsf{P} & & \\
\cap\!| & & \cap\!| & & \\
\mathrm{FO}[2^{n^{O(1)}}] & = & \mathsf{PSPACE} & = & \mathrm{SO}[n^{O(1)}]
\end{array}
$$

Figure 3.1. Iterated quantifier block logics and complexity classes.

3.2 Succinctness Bounds

To prove lower bounds on the size of formulas, we need a more refined version of Ehrenfeucht-Fraïssé games to take into account the size of the game tree instead of just its depth. Adler-Immerman games were introduced in [1] for exactly this purpose.

Definition 3.2.1 (Adler-Immerman Game)**.** [1] The *Adler-Immerman game* on two sets of structures $\mathcal{A}$ and $\mathcal{B}$ is a game with two players Samson and Delilah. During the course of the game, a tree is constructed where each node is labeled with a pair of sets of structures. Samson moves first, and Delilah can respond to some of Samson's moves.

- Initially, the tree consists only of a root node with label $(\mathcal{A}, \mathcal{B})$.
- Samson can close a leaf node of his choice in one move if there is an atomic formula that is satisfied by all structures from the left set of the pair, but not satisfied by any structure from the right set. After closing the leaf node, no further moves on it are possible.
- Samson can play one of the following moves on an open leaf node with label $(\mathcal{A}_0, \mathcal{B}_0)$.
  - In a NOT move, he adds a new child node with label $(\mathcal{B}_0, \mathcal{A}_0)$.
  - In an OR move, he picks two sets $\mathcal{A}_1, \mathcal{A}_2 \subseteq \mathcal{A}_0$ such that $\mathcal{A}_1 \cup \mathcal{A}_2 = \mathcal{A}_0$, and attaches two child nodes to the current node with labels $(\mathcal{A}_1, \mathcal{B}_0)$ and $(\mathcal{A}_2, \mathcal{B}_0)$.
  - In an existential move on a variable x, Samson picks one designated element $s(A)$ for every structure $A \in \mathcal{A}_0$. Delilah responds by picking a set of elements $D(B)$ for every structure $B \in \mathcal{B}_0$. A new child node with label

    $$(\{(A, s(A)) \mid A \in \mathcal{A}_0\}, \{(B, i) \mid B \in \mathcal{B}_0, i \in D(B)\})$$

    is attached to the current node.
- Samson wins the game if he can close all leaves, otherwise Delilah wins.

When restricted to boolean moves only, the Adler-Immerman game is exactly the Karchmer-Wigderson communication complexity game [25]. If we restrict the game to only NOT and existential moves, and also require that $|D(B)| = 1$ and $|\mathcal{A}| = |\mathcal{B}| = 1$, then this game is just a standard Ehrenfeucht-Fraïssé game. We note that Delilah has an obvious optimal strategy in every Adler-Immerman game: Pick $D(B) = B$ in every existential move. Extensions of this game for temporal logics and transitive closure operators are relatively straightforward to define.

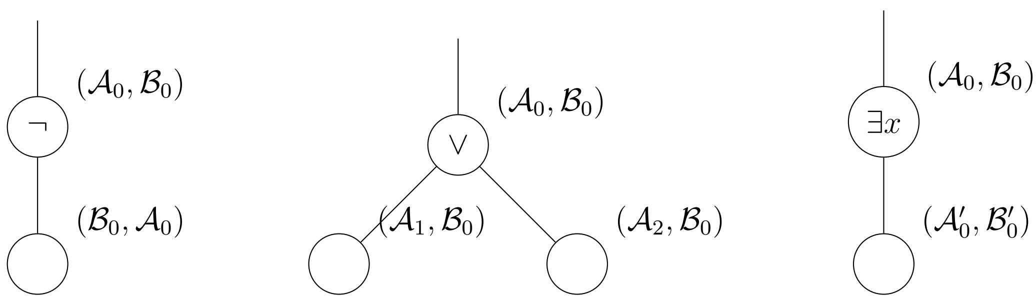

Figure 3.2. Moves in the Adler-Immerman Game.

Theorem 3.2.2 (Fundamental Theorem of Adler-Immerman Games)**.** [1] Samson has a winning strategy for the m-move k-variable Adler-Immerman game on a pair of sets of structures $(\mathcal{A}, \mathcal{B})$ if and only if there is a FOSIZE$^k[m]$ formula that distinguishes between every structure in $\mathcal{A}$ and every structure in $\mathcal{B}$.

After their introduction in [1], Adler-Immerman games have been used successfully to prove interesting bounds on the succinctness of the finite-variable fragments of first-order logic on linear orders [15]: The succinctness of FO^3 in FO^2 is $O(n^4)$, and the succinctness of both FO and FO^4 in FO^3 is $2^{\Theta(n)}$. Linear orders are about the weakest class of structures one could think of, but all lower bounds translate directly to more complicated classes of structures like finite words and graphs, as long as an ordering relation is present. Upper bounds generally do not translate as easily.

Two interesting questions left open, the first one of which we feel confident enough about to formulate as a conjecture, are the following.

Conjecture 3.2.3. The succinctness of FO^3 in FO^2 on linear orders is $\Theta(n^2)$.

Open Problem 3.2.4. For any $k \geq 4$, determine the succinctness of FO^{k+1} in FO^k on linear orders.

To prove lower bounds with Adler-Immerman games, the following two techniques have been established.

- Incompatible Pairs Technique [1]: Two pairs of structures (A_1, B_1) and (A_2, B_2) are *incompatible* if they need to appear separately on at least one leaf. In other words, Samson cannot win the game on just $(\{A_1, A_2\}, \{B_1, B_2\})$ without completely separating either A_1 from A_2 or B_1 from B_2 on at least one branch of the tree. The number of mutually incompatible pairs yields a lower bound on the number of leaves, and thus on the size of the tree.

- Weight Function Technique [15]: Define a function $w : \mathcal{P}(\mathcal{A}) \times \mathcal{P}(\mathcal{B}) \to \mathbb{R}$, argue bottom-up about the maximum increase of w in each move, and bound $w(\mathcal{A}, \mathcal{B})$.

The weight function technique is used in [15] to prove bounds for expressing properties of linear orders. Of particular interest are the following two properties.

$$\mathsf{LENGTH}_n := \{\mathrm{LO}_i \mid i \leq n\}$$

$$\mathsf{EVEN\text{–}LENGTH}_n := \{\mathrm{LO}_i \mid i \leq n \text{ and } i \text{ is even}\}$$

Definition 3.2.5. [15] A *separator* for two sets of structures $\mathcal{A}$ and $\mathcal{B}$ is a function

$$\delta : \mathcal{P}_2(\{\min, x, y, z, \max\}) \to \mathbb{N}$$

such that for all $A \in \mathcal{A}$, $B \in \mathcal{B}$, there are $u, v \in \{\min, x, y, z, \max\}$ with

- $\mathrm{ord}(u^A, v^A) \neq \mathrm{ord}(u^B, v^B)$ and $\delta(\{u, v\}) > 0$, or

- $\mathrm{dist}(u^A, v^A) \neq \mathrm{dist}(u^B, v^B)$ and $\delta(\{u, v\}) \geq \min\{\mathrm{dist}(u^A, \hat{y}^A), \mathrm{dist}(u^B, v^B)\}$.

Intuitively, a separator gives a bound on the distance required to walk on two structures in order to distinguish them. Based on separators, a weight function is defined depending on a minimal separator for $(\mathcal{A}, \mathcal{B})$, involving summing up several distances and taking square roots. Using this weight function in an elaborate case-distinction, it is then proved in [15] that any winning strategy for Samson in the three-variable Adler-Immerman game on $(\{\mathsf{LENGTH}_n\}, \{\mathsf{LENGTH}_{n+1}\})$ requires a tree size of $\Omega(\sqrt{n})$, and thus that $\mathsf{LENGTH}_n \notin \mathrm{FOSIZE}^3[o(\sqrt{n})]$. Their succinctness bounds are an immediate consequence of this lower bound. We conjecture that this lower bound is in fact linear and matches the trivial upper bound.

Conjecture 3.2.6. $\mathsf{LENGTH}_n \notin \mathrm{FOSIZE}^3[o(n)]$.

An immediate consequence of this conjecture would be that the succinctness of FO^3 in FO^2 is $O(n^2)$, a first step towards proving Conjecture 3.2.3. The following proposition summarizes our current knowledge on upper bounds for expressing LENGTH_n in first-order logic.

Proposition 3.2.7. $\mathsf{LENGTH}_n \in \mathrm{FOSIZE}^2[O(n)] \cap \mathrm{FO}^3_{O(\log n)} \cap \mathrm{FOSIZE}^4[O(\log n)]$.

Proof. For expressing LENGTH_n with two variables, we can simply walk over the structure from left to right with the following recursively defined formula. This places LENGTH_n in $\mathrm{FO}^2[n]$ and thus $\mathrm{FOSIZE}^2[O(n)]$.

$$\mathrm{length}^2_{\geq n}(x) := \exists y.(y > x)\, \mathrm{length}^2_{\geq n-1}(y)$$
$$\mathrm{length}^2_{\geq 1}(x) := x \leq \max$$

With three variables, we can reduce the quantifier depth, but we do not know if the size of the formula can be reduced.

$$\text{length}^3_{\geq n}(x,y) := \exists z \left(\text{length}^3_{\geq\lceil n/2\rceil}(x,z) \wedge \text{length}^3_{\geq\lfloor n/2\rfloor}(z,y)\right)$$
$$\text{length}^3_{\geq 1}(x,y) := x < y$$

With four variables we can place LENGTH_n in $\mathrm{FO}^4[\log n]$ and thus $\mathrm{FOSIZE}^4[O(\log n)]$ if n is a power of two. For other numbers, the size bound still holds, but the quantifier blocks differ according to the bit at the corresponding position of the binary expansion of n. Here, we only present the version with uniform quantifier blocks. The universal quantifier on the fourth variable is only used to unify the variable names for the recursion.

$$\text{length}^4_{\geq n}(x,y) := \exists z.(x < z < y \vee x > z > y)\, \forall v.(v = x \vee v = y)\, \text{length}^4_{\geq n/2}(v,z)$$
$$\text{length}^4_{\geq 1}(x,y) = x \neq y \qquad \square$$

It should be noted that the four-variable bound is optimal since a standard Ehrenfeucht-Fraïssé game argument shows that $\mathsf{LENGTH}_n \notin \mathrm{FO}_{o(\log n)}$. For further progress on Conjecture 3.2.3, we believe that the properties $\mathsf{EVEN}\text{–}\mathsf{LENGTH}_n$ will be crucial. As before, we first state some known upper bounds.

Proposition 3.2.8. $\mathsf{EVEN}\text{–}\mathsf{LENGTH}_n \in \mathrm{FOSIZE}^2[O(n^2)] \cap \mathrm{FOSIZE}^3[O(n)]$.

Proof. For the first bound, we take the disjunction over all linear orders of even length up to n.

$$\text{even}^2_n := \bigvee\nolimits_{i \leq n \text{ and } i \text{ even}} \text{length}^2_{=i}$$
$$\text{length}^2_{=n} := \text{length}^2_{\geq n} \wedge \neg\text{length}^2_{\geq n+1}$$

For the three-variable bound, we use that fact that any even number can be written as the sum of two even numbers that are close to half of the original number.

Without loss of generality we assume that n is even.

$$\mathrm{even}^3_n(x,y) := \exists z.(x<z<y)\ \left(\mathrm{even}^3_{2\lceil n/4\rceil}(x,z) \wedge \mathrm{even}^3_{2\lfloor n/4\rfloor}(z,y)\right)$$
$$\mathrm{even}^3_2(x,y) := \exists z\,(\mathrm{Suc}(x,z)\wedge \mathrm{Suc}(z,y)) \vee x=y \qquad \square$$

In order to completely settle Conjecture 3.2.3, all that is needed is to prove a lower bound that matches the stated upper bound on $\mathsf{EVEN\text{–}LENGTH}_n$ for FO^2.

3.3 A Simple Lower Bound

We present a new lower bound on formula size using the separator technique. While this lower bound does not settle Conjecture 3.2.3, we hope that an extension of our construction will succeed in that goal.

Proposition 3.3.1. Any $\mathrm{FO}^2[<]$ formula that distinguishes between LO_n and LO_{n+1} has quantifier depth at least $\lceil n/2\rceil + 1$.

Proof. We use our results on the structure of FO^2, in particular Theorem 1.1.8. LO_n and LO_{n+1} agree on the ordering of all $\lceil n/2\rceil$-rankers with respect to all $\lfloor n/2\rfloor$-rankers. Thus any formula that distinguishes between the two structures has quantifier depth greater than $\lceil n/2\rceil$. $\square$

This bound is the best we can hope for using our structure theorem, since the two linear orders disagree on the ordering of the two rankers

$$r = \triangleright^{\lceil n/2\rceil+1} \qquad \text{and} \qquad s = \triangleleft^{\lceil n/2\rceil} \quad .$$

A tighter lower bound can be attained using the separator method.

Proposition 3.3.2. Any FO^2 formula that distinguishes between LO_n and LO_{n+1} has size at least $n-1$.

Definition 3.3.3. Let A and B be linear orders, and let $z \in \{x, y\}$ and $m \in \{\min, \max\}$. The (z, m)-*separation distance* of A and B is

$$\sigma_{z,m}(A, B) := \begin{cases} 0 & \text{if } A \text{ and } B \text{ disagree on } z = m \\ \infty & \text{if } \mathrm{dist}_A(z, m) = \mathrm{dist}_B(z, m) \\ \min\{\mathrm{dist}_A(z, m), \mathrm{dist}_B(z, m)\} & \text{otherwise} \end{cases}$$

We also define

$$\begin{aligned} \sigma_z(A, B) &:= \min\{\sigma_{z,\min}(A, B), \sigma_{z,\max}(A, B)\} \\ \sigma_m(A, B) &:= \min\{\sigma_{x,m}(A, B), \sigma_{y,m}(A, B)\} \\ \sigma(A, B) &:= \min\{\sigma_x(A, B), \sigma_y(A, B)\} \quad . \end{aligned}$$

Definition 3.3.4. Let $\mathcal{A}$ and $\mathcal{B}$ be two sets of structures. A *2-variable line separator* for $\mathcal{A}$ and $\mathcal{B}$ is a mapping

$$\delta : \{\{\min, x\}, \{\min, y\}, \{x, \max\}, \{y, \max\}\} \to \mathbb{N}$$

such that for every pair of structures $A \in \mathcal{A}$ and $B \in \mathcal{B}$, at least one of the following conditions holds.

(a) A and B disagree on an atomic predicate.

(b) For some $v \in \{x, y\}$ and some $m \in \{\min, \max\}$, $\mathrm{dist}_A(v, m) \neq \mathrm{dist}_B(v, m)$ and $\min\{\mathrm{dist}_A(v, m), \mathrm{dist}_B(v, m)\} \leq \delta(\{v, m\})$.

For simplicity we write $\delta(v, m)$ instead of $\delta(\{v, m\})$, and we generally order the arguments as they occur on the structures.

Definition 3.3.5. Let $\mathcal{A}$ and $\mathcal{B}$ be two sets of structures. The *cost* of a 2-variable line separator δ is

$$\mathrm{cost}(\delta) := \delta(\min, x) + \delta(\min, y) + \delta(x, \max) + \delta(y, \max) \quad .$$

The *2-variable line separation cost* of $\mathcal{A}$ and $\mathcal{B}$ is

$$\rho(\mathcal{A}, \mathcal{B}) := \min\{\mathrm{cost}(\delta) \mid \delta \text{ is a line separator for } \mathcal{A} \text{ and } \mathcal{B}\} \quad .$$

For the rest of this section, we will refer to 2-variable line separators simply as *line separators.*

To prove Proposition 3.3.2, we argue that the line separation cost after the first quantifier move is n, and we show that quantifier moves decrease the line separation cost by at most one. In the following sequence of lemmas, we look at what happens to the line separation cost throughout the Adler-Immerman game played on the two structures.

Lemma 3.3.6 (OR move)**.** Let $\mathcal{A}$ and $\mathcal{B}$ be sets of linear orders, and suppose that $\mathcal{A} = \mathcal{A}' \cup \mathcal{A}''$. Then $\rho(\mathcal{A}, \mathcal{B}) \leq \rho(\mathcal{A}', \mathcal{B}) + \rho(\mathcal{A}'', \mathcal{B})$.

Proof. Let δ_1 and δ_2 be line separators of minimal separation cost for $(\mathcal{A}_1, \mathcal{B})$ and $(\mathcal{A}_2, \mathcal{B})$, respectively. We define the line separator δ for $(\mathcal{A}, \mathcal{B})$ as the point-wise maximum of δ_1 and δ_2. Obviously δ is a line separator for $(\mathcal{A}, \mathcal{B})$, and we have

$$\begin{aligned}
\mathrm{cost}(\delta) = \max\{&\delta_1(\min, x), \delta_2(\min, x)\} + \max\{\delta_1(\min, y), \delta_2(\min, y)\} \\
&+ \max\{\delta_1(x, \max), \delta_2(x, \max)\} + \max\{\delta_1(y, \max), \delta_2(y, \max)\} \\
\leq\ &\delta_1(\min, x) + \delta_2(\min, x) + \delta_1(\min, y) + \delta_2(\min, y) \\
&+ \delta_1(x, \max) + \delta_2(x, \max) + \delta_1(y, \max) + \delta_2(y, \max) \\
=\ &\mathrm{cost}(\delta_1) + \mathrm{cost}(\delta_2)
\end{aligned}$$

$\square$

The separation cost is not affected by NOT moves since $\rho(\mathcal{A}, \mathcal{B}) = \rho(\mathcal{B}, \mathcal{A})$.

The following lemma is a crucial part of our analysis of existential moves. While the lemma is formulated for an existential move on the variable x, an analogous version holds for existential moves on the variable y.

Lemma 3.3.7 (Existential move on x on a pair of linear orders)**.** Let A and B be linear orders of different size that agree on all atomic formulas, let $f(A) \in |A|$, and set

$$\mathcal{A}' := \{A'\} \quad \text{where} \quad A' := (A, f(A)) \quad \text{and} \quad \mathcal{B}' := \{(B, b) \mid b \in |B|\} \quad .$$

Then one of the following three statements holds.

(a) $\mathrm{dist}_B(\min, y) \le \mathrm{dist}_{A'}(\min, x) < \mathrm{dist}_A(\min, y)$ and there is a structure $B' \in \mathcal{B}'$ such that $\sigma_{x,\min}(A', B') = \sigma_{y,\min}(A, B) - 1$ and

$$\sigma_{x,\max}(A', B') > \min\{\mathrm{dist}_A(y, \max), \mathrm{dist}_B(y, \max)\} \quad .$$

(b) $\mathrm{dist}_B(y, \max) \le \mathrm{dist}_{A'}(x, \max) < \mathrm{dist}_A(y, \max)$ and there is a structure $B' \in \mathcal{B}'$ such that $\sigma_{x,\max}(A', B') = \sigma_{y,\max}(A, B) - 1$ and

$$\sigma_{x,\min}(A', B') > \min\{\mathrm{dist}_A(\min, y), \mathrm{dist}_B(\min, y)\} \quad .$$

(c) There is $B' \in \mathcal{B}'$ such that $\sigma_{x,\min}(A', B') \ge \min\{\mathrm{dist}_A(y, \min), \mathrm{dist}_B(y, \min)\}$ and $\sigma_{x,\max}(A', B') \ge \min\{\mathrm{dist}_A(y, \max), \mathrm{dist}_B(y, \max)\}$.

Samson is only making progress at separating the two structures if (a) or (b) apply.

Proof. If Samson places x at the position of y, i.e. $f(A) = y^A$, then we choose $B' := (B, y^B)$, and (c) holds. Otherwise we have $f(A) < y^A$ or $f(A) > y^A$. Since both of these cases are symmetric (with statements (a) and (b) interchanged), we only consider the case where $f(A) < y^A$, and thus $\mathrm{dist}_{A'}(\min, x) < \mathrm{dist}_A(\min, y)$. We observe that since $y^A > 1$ and A and B agree on all atomic formulas, it must also be the case that $y^B > 1$.

We start with the case where $\mathrm{dist}_B(\min, y) \leq \mathrm{dist}_{A'}(\min, x) < \mathrm{dist}_A(\min, y)$. This situation is illustrated in Figure 3.3. We choose $B' = (B, y^B - 1)$. If $y^B - 1 = \min^B$,

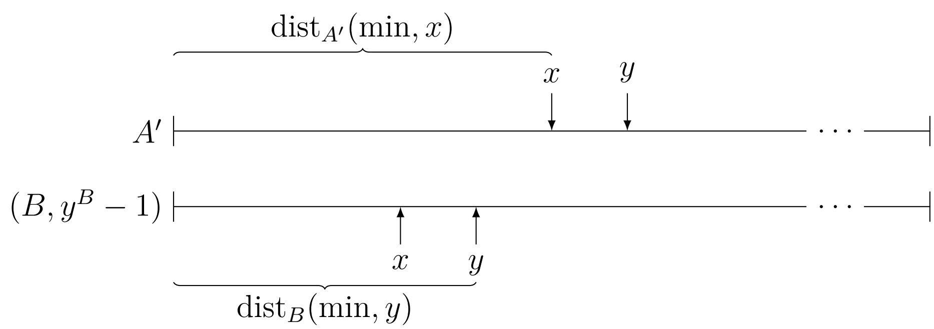

Figure 3.3. Proof of Proposition 3.3.7: Samson makes progress by placing x on A in between y^B and y^A.

then A' and B' disagree on an atomic formula and $\sigma_{x,\min}(A', B') = 0$. However, in this case we also have $\sigma_{y,\min}(A', B') \leq 1$, and thus (a) holds. Otherwise A' and B' agree on all atomic formulas. Since $\mathrm{dist}_{A'}(\min, x) > \mathrm{dist}_{B'}(\min, x)$, we have

$$\sigma_{x,\min}(A', B') = \mathrm{dist}_{B'}(\min, x) = \mathrm{dist}_B(\min, y) - 1 = \sigma_{y,\min}(A, B) - 1 \quad .$$

Furthermore,

$$\begin{aligned} \sigma_{x,\max}(A', B') &\geq \min\{\mathrm{dist}_{A'}(x, \max), \mathrm{dist}_{(B, y^B - 1)}(x, \max)\} \\ &> \min\{\mathrm{dist}_A(y, \max), \mathrm{dist}_B(y, \max)\} \quad . \end{aligned}$$

Thus (a) holds.

Otherwise we have $\text{dist}_{A'}(\min, x) < \text{dist}_B(\min, y)$, and we choose $B' = (B, f(A))$, as illustrated in Figure 3.4. The pair (A', B') agrees atomically, and $\text{dist}_{A'}(\min, x) =$

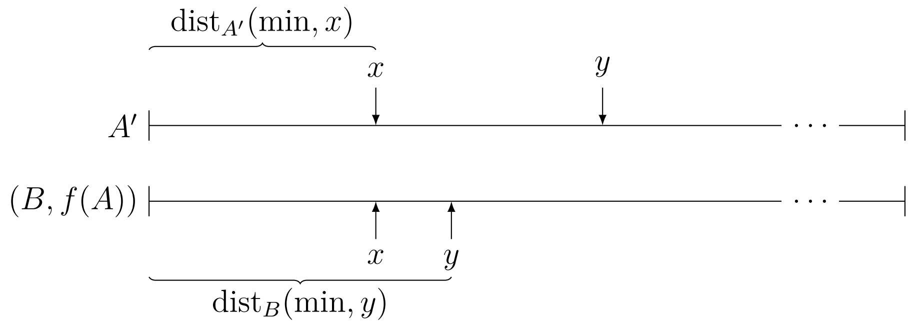

Figure 3.4. Proof of Proposition 3.3.7: Samson places x too far to the left of y, and thus he makes no progress.

$\text{dist}_{B'}(\min, x)$. Thus $\sigma_{x,\min}(A', B') = \infty$, and

$$\begin{aligned}\sigma_{x,\max}(A', B') &= \min\{\text{dist}_{A'}(x, \max), \text{dist}_{B'}(x, \max)\} \\ &> \min\{\text{dist}_A(y, \max), \text{dist}_B(y, \max)\} \quad .\end{aligned}$$

This establishes (c). ☐

As with the previous lemma, we formulate the following lemma for an existential move on the variable x, but an analogous version holds for existential moves on the variable y.

Lemma 3.3.8 (Existential move on x)**.** Let $\mathcal{A}$ and $\mathcal{B}$ be sets of linear orders, let n be the minimum size of all structures in $\mathcal{A}$ and $\mathcal{B}$, and let f be a function that maps any linear order $A \in \mathcal{A}$ to an element of $|A|$. We define the two sets of structures

$$\mathcal{A}' := \{(A, f(A)) \mid A \in \mathcal{A}\} \quad \text{and} \quad \mathcal{B}' := \{(B, b) \mid B \in \mathcal{B}, b \in |B|\} \quad .$$

If $\rho(\mathcal{A}, \mathcal{B}) \geq 1$ then $\rho(\mathcal{A}', \mathcal{B}') \geq \min\{\rho(\mathcal{A}, \mathcal{B}) - 1, n - 3\}$.

Proof. Let δ be a line separator for $(\mathcal{A}, \mathcal{B})$ of minimal cost, under the constraint that $\delta(\min, x) = \delta(x, \max) = 0$. We have $\mathrm{cost}(\delta) \geq \rho(\mathcal{A}, \mathcal{B})$. We use the minimality of δ to identify pairs of linear orders that are hard to separate, and then argue that separating these linear orders is still hard after the quantifier move.

Let $\ell := \delta(\min, y)$ and let $r := \delta(y, \max)$. We always have $\ell + r > 0$ since $\rho(\mathcal{A}, \mathcal{B}) \geq 1$. To argue that $\rho(\mathcal{A}', \mathcal{B}') \geq \min\{\rho(\mathcal{A}, \mathcal{B}) - 1, n - 3\}$, we let δ' be an arbitrary line separator for $(\mathcal{A}', \mathcal{B}')$, and let $\ell' := \delta'(\min, x)$ and $r' := \delta'(x, \max)$.

The outline for the remainder of this proof is as follows: We first consider the case where $\ell = 0$. The case for $r = 0$ is completely symmetric

For the sake of a contradiction, suppose that δ' is a line separator for $(\mathcal{A}', \mathcal{B}')$ with $\mathrm{cost}(\delta') < \min\{\ell + r - 1, n - 3\}$. We set $\ell' := \max\{\delta'(\min, x), \delta'(\min, y)\}$ and $r' := \max\{\delta'(x, \max), \delta'(y, \max)\}$, thus $\ell' + r' \leq \mathrm{cost}(\delta') < \min\{\ell + r - 1, n - 3\}$, and $\ell' + r' < \ell + r - 1$. We first consider the case where both $\ell' < \ell$ and $r' < r$, and then the case where $\ell' \geq \ell$ and $r' < r - 1$. The case where $r' \geq r$ and $\ell' < \ell - 1$ is completely symmetric.

In the case where $\ell' < \ell$ and $r' < r$, we use the minimality of δ to find linear orders $C, E \in \mathcal{A}$ and $D, F \in \mathcal{B}$ such that separating the pair (C, D) requires walking ℓ steps to the left and separating (E, F) requires walking r steps to the right. More formally, C, D, E and F satisfy all of the following conditions.

- C and D agree on all atomic formulas.
- $\min\{\mathrm{dist}_C(\min, y), \mathrm{dist}_D(\min, y)\} = \ell$ and $\mathrm{dist}_C(\min, y) \neq \mathrm{dist}_D(\min, y)$.
- $\min\{\mathrm{dist}_C(y, \max), \mathrm{dist}_D(y, \max)\} > r$ or $\mathrm{dist}_C(y, \max) = \mathrm{dist}_D(y, \max)$.
- E and F agree on all atomic formulas.
- $\min\{\mathrm{dist}_E(y, \max), \mathrm{dist}_F(y, \max)\} = r$ and $\mathrm{dist}_E(y, \max) \neq \mathrm{dist}_F(y, \max)$.
- $\min\{\mathrm{dist}_E(\min, y), \mathrm{dist}_F(\min, y)\} > \ell$ or $\mathrm{dist}_E(\min, y) = \mathrm{dist}_F(\min, y)$.

Using the notation from Lemma 3.3.7, let $\mathcal{C}'$ and $\mathcal{D}'$ be the sets of structures after the existential move on (C, D), and let $\mathcal{E}'$ and $\mathcal{F}'$ be the sets of structures after the existential move on (E, F). The pair $(\mathcal{C}', \mathcal{D}')$ cannot be separated by δ' on y. We apply Lemma 3.3.7 to the pair (C, D), and see that if statements (b) or (c) from that lemma apply, then δ' cannot separate $(\mathcal{C}', \mathcal{D}')$. Thus part (a) has to apply, and we have $\delta'(\min, x) \geq \ell - 1$. Similarly, the pair $(\mathcal{E}', \mathcal{F}')$ cannot be separated by δ' on y. We apply Lemma 3.3.7 to the pair (E, F), and see that if statements (a) or (c) from that lemma apply, then δ' cannot separate $(\mathcal{E}', \mathcal{F}')$. Thus part (b) has to apply, and we have $\delta'(x, \max) \geq r - 1$. All together, the situation illustrated in Figure 3.5 applies.

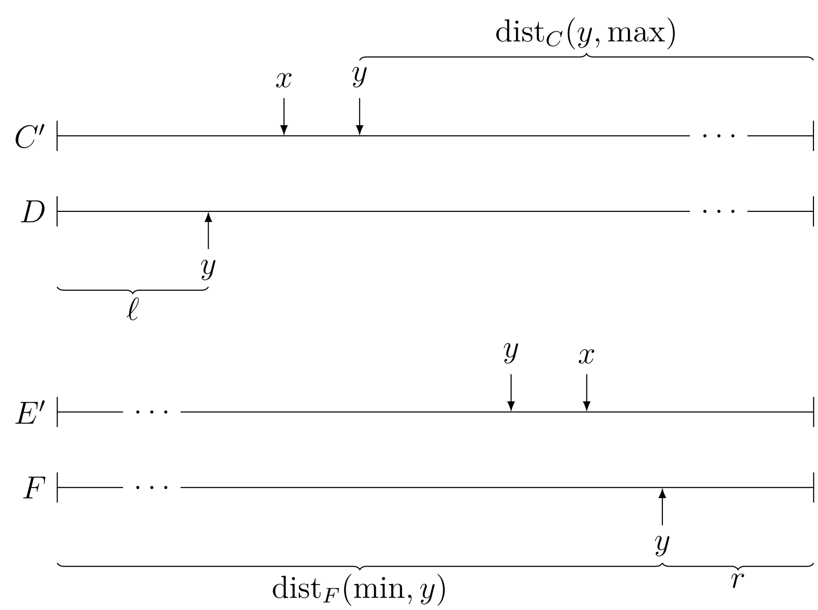

Figure 3.5. Proof of Lemma 3.3.8: Four structures in the case where $\ell' < \ell$ and $r' < r$.

But δ' also needs to separate the pair (C', F'_{y^F-1}), and this separation, as above, cannot be on y. To show that this pair also cannot be separated on $(\min, x)$ nor on $(x, \max)$, we consider whether $\mathrm{dist}_C(y, \max) = \mathrm{dist}_D(y, \max)$ or $\mathrm{dist}_E(\min, y) = \mathrm{dist}_F(\min, y)$. There are four cases.

Case 1: $\text{dist}_C(y, \max) \neq \text{dist}_D(y, \max)$ and $\text{dist}_E(\min, y) \neq \text{dist}_F(\min, y)$. We have $\text{dist}_C(y, \max) > r$ and $\text{dist}_F(\min, y) > \ell$. If δ' separates this pair on $(\min, x)$, then $\delta'(\min, x) \geq \ell$, and if δ' separates on $(x, \max)$ then $\delta'(x, \max) > r$. Thus δ' cannot be a separator for (C', F'_{y^F-1}).

Case 2: $\text{dist}_C(y, \max) = \text{dist}_D(y, \max)$ and $\text{dist}_E(\min, y) \neq \text{dist}_F(\min, y)$. We have $\text{dist}_F(\min, y) > \ell$. If δ' separates (C', F'_{y^F-1}) on $(\min, x)$, then $\delta'(\min, x) \geq \ell$, a contradiction. If δ' separates on $(x, \max)$, then

$$\delta'(x, \max) \geq \min\{\text{dist}_{C'}(x, \max), \text{dist}_{F'_{y^F-1}}(x, \max)\} \geq \min\{\text{dist}_{C'}(x, \max), r + 1\} \quad .$$

But since $\delta'(x, \max) < r$, it must be the case that $\delta'(x, \max) \geq \text{dist}_{C'}(x, \max)$. Then

$$\begin{aligned}
\text{cost}(\delta') \geq \delta'(\min, x) + \delta'(x, \max) &\geq \ell - 1 + \text{dist}_{C'}(x, \max) \\
&\geq \ell - 1 + \text{dist}_C(y, \max) + 1 \\
&= \text{dist}_D(\min, y) + \text{dist}_D(y, \max) \\
&= \|D\| - 1 \geq n - 1 \quad .
\end{aligned}$$

Case 3: $\text{dist}_C(y, \max) \neq \text{dist}_D(y, \max)$ and $\text{dist}_E(\min, y) = \text{dist}_F(\min, y)$. This is completely symmetric to the previous case on the pair of structures (E, D).

Case 4: $\text{dist}_C(y, \max) = \text{dist}_D(y, \max)$ and $\text{dist}_E(\min, y) = \text{dist}_F(\min, y)$. If δ' separates the pair (C', F'_{y^F-1}) on $(x, \max)$, we use the second part of the argument from case 2 to see that $\text{cost}(\delta') \geq n - 1$. Otherwise the pair is separated on $(\min, x)$, and

$$\delta'(\min, x) \geq \min\{\text{dist}_{C'}(\min, x), \text{dist}_{F'_{y^F-1}}(\min, x)\} \geq \min\{\ell, \text{dist}_{F'_{y^F-1}}(\min, x)\} \quad .$$

But since $\delta'(\min, x) < \ell$, it must be the case that $\delta'(\min, x) \geq \text{dist}_{F'_{y^F-1}}(\min, x)$. Then

$$
\begin{aligned}
\mathrm{cost}(\delta') \geq \delta'(\min, x) + \delta'(\min, x) &\geq \mathrm{dist}_{F'_{y^F-1}}(\min, x) + r - 1 \\
&= \mathrm{dist}_F(\min, y) - 1 + \mathrm{dist}_F(y, \max) - 1 \\
&= \|F\| - 3 \geq n - 3 \quad .
\end{aligned}
$$

Now we consider the case where $\ell' \geq \ell$ and $r' < r - 1$. Since δ is minimal and $\mathrm{cost}(\delta) = \ell + r > \ell' + r' + 1$, there are structures $C \in \mathcal{A}$ and $D \in \mathcal{B}$ such that all of the following three statements hold.

- C and D agree on all atomic formulas.
- $\min\{\mathrm{dist}_C(\min, y), \mathrm{dist}_D(\min, y)\} > \ell'$ or $\mathrm{dist}_C(\min, y) = \mathrm{dist}_D(\min, y)$.
- $\min\{\mathrm{dist}_C(y, \max), \mathrm{dist}_D(y, \max)\} > r' + 1$ or $\mathrm{dist}_C(y, \max) = \mathrm{dist}_D(y, \max)$.

Furthermore, since δ separates (C, D) on y, it needs to separate (C, D) on $(y, \max)$ and thus $\mathrm{dist}_C(y, \max) \neq \mathrm{dist}_D(y, \max)$, therefore $\min\{\mathrm{dist}_C(y, \max), \mathrm{dist}_D(y, \max)\} > r' + 1$.

Similarly, there are structures $E \in \mathcal{A}$ and $F \in \mathcal{B}$ such that all of the following three statements hold.

- E and F agree on all atomic formulas.
- $\min\{\mathrm{dist}_E(\min, y), \mathrm{dist}_F(\min, y)\} > \ell' + 1$ or $\mathrm{dist}_E(\min, y) = \mathrm{dist}_F(\min, y)$.
- $\min\{\mathrm{dist}_E(y, \max), \mathrm{dist}_F(y, \max)\} > r'$ or $\mathrm{dist}_E(y, \max) = \mathrm{dist}_F(y, \max)$.

Furthermore, since δ separates (E, F) on y, it needs to separate on $(\min, y)$ and thus $\mathrm{dist}_E(\min, y) \neq \mathrm{dist}_F(\min, y)$, therefore $\min\{\mathrm{dist}_E(\min, y), \mathrm{dist}_F(\min, y)\} > \ell' + 1$.

Applying Lemma 3.3.7 to (C, D), we see that if conditions (b) or (c) apply, then δ' cannot separate $(\mathcal{C}', \mathcal{D}')$. Thus (c) has to be true for this pair. Applying the same lemma to (E, F), we see that δ' cannot separate $(\mathcal{E}', \mathcal{F}')$ if statements (b) or (c) hold, thus (a) has to be true. Hence the situation illustrated in Figure 3.6 applies.

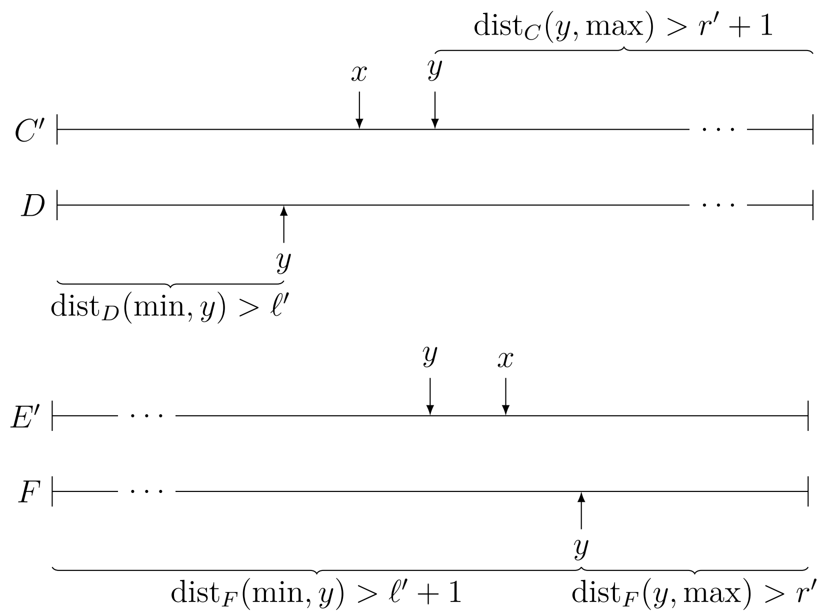

Figure 3.6. Proof of Lemma 3.3.8: Four structures in the case where $\ell' \geq \ell$ and $r' < r - 1$.

The pair (C', F'_{y^F-1}) cannot be separated by δ' on either y nor x, a contradiction.

$\square$

Proof of Proposition 3.3.2. We consider the situation after the first quantifier move in the game on LO_n and LO_{n+1}. If Samson placed his first pebble on LO_n at position i, then Delilah's reply includes the structures (LO_{n+1}, i) and $(\mathrm{LO}_{n+1}, i+1)$. Thus the line separation cost for this pair of sets of structures is at least $(n+1) - i + i = n+1$. If Samson placed his first pebble on LO_{n+1} at position i, then Delilah's reply includes the structures (LO_n, i) and $(\mathrm{LO}_n, i-1)$, leading to a line separation cost of at least $n - i + i - 2 = n - 2$.

Looking at the game tree of any FO^2 formula that separates these two linear orders, we notice that all the children have separators of weight 0 since they disagree on atomic formulas. The weight of minimal separators increases from bottom to top as stated in Lemma 3.3.6 for OR moves and Lemma 3.3.8 for existential moves. Thus to have a separator of weight $n-2$ at the top we need at least $n-2$ quantifier nodes in the tree.

Including the first quantifier move, the total number of quantifier moves is at least $n-1$. □

3.4 Towards Settling Our First Conjecture

To settle Conjecture 3.2.6, we propose to bound the formula size required to separate the following two sets of linear orders.

$$\mathcal{A}_n := \{\mathrm{LO}_i \mid n \leq i \leq 2n, i \text{ even}\} \qquad \mathcal{B}_n := \{\mathrm{LO}_i \mid n \leq i \leq 2n, i \text{ odd}\}$$

After the first pair of pebbles is placed in the initial move, there are about n different distances to walk towards the left or the right, and it does not seem possible for Samson to make concurrent and sustained progress on more than one of these distances at a time. Thus we conjecture that a quadratic number of moves is required.

Conjecture 3.4.1. Any $\mathrm{FO}^2[<]$ formula that distinguishes between $\mathcal{A}_n$ and $\mathcal{B}_n$ has size $\Omega(n^2)$.

To prove this conjecture, we need more refined separators, since our earlier separators (Definition 3.3.4) can separate $\mathcal{A}_n$ and $\mathcal{B}_n$ with cost $2n$. The following definition takes into account that when Samson decides to walk to the left on one set of structures, he can only make progress on the structures where to distance he needs to walk is larger than the corresponding distances in the other set of structures.

Definition 3.4.2. Let $\mathcal{A}$ and $\mathcal{B}$ be sets of linear orders, and let $d \in \mathbb{N}$, $z \in \{x, y\}$, $m \in \{\min, \max\}$. The pair $(\mathcal{A}, \mathcal{B})$ is (d, z, m)-*separated* if there is $\sim \in \{<, >\}$ such that for all $A \in \mathcal{A}$ and all $B \in \mathcal{B}$,

$$\mathrm{dist}_A(z, m) \sim \mathrm{dist}_B(z, m) \text{ and } \sigma_{z,m}(A, B) \leq d.$$

When not all three elements of the tuple (d, z, m) are specified, existential quantification is implied. For example, $(\mathcal{A}, \mathcal{B})$ is *d-separated* if there are z and m such that $(\mathcal{A}, \mathcal{B})$ is (d, z, m)-separated.

Definition 3.4.3. The multiset $\delta = \{d_1, \ldots, d_k\}$ with $k \in \mathbb{N}$ and $d_i \in \mathbb{N}$ for $1 \leq i \leq k$ *separates* the pair of sets of structures $(\mathcal{A}, \mathcal{B})$ if there are sets $\mathcal{A}_1, \ldots, \mathcal{A}_k \subseteq \mathcal{A}$ and $\mathcal{B}_1, \ldots, \mathcal{B}_k \subseteq \mathcal{B}$ such that $\bigcup_{i=1}^{k} (\mathcal{A}_i \times \mathcal{B}_i) = \mathcal{A} \times \mathcal{B}$ and for all $1 \leq i \leq k$, $(\mathcal{A}_i, \mathcal{B}_i)$ is d_i-separated. The *cost* of the multiset δ is $\text{cost}(\delta) := \sum_{i=1}^{k} d_i$. The *minimum separation cost* of $(\mathcal{A}, \mathcal{B})$, $\rho(\mathcal{A}, \mathcal{B})$, is the minimum cost of a multiset that separates $(\mathcal{A}, \mathcal{B})$.

As above, the analysis for OR moves is simple, but proving a meaningful lemma for existential moves is a challenge.

CHAPTER 4

GENERALIZED STAR-HEIGHT

Generalized regular expressions are the natural extension of common regular expressions with a complementation operator. To avoid confusion, we generally refer to the standard notion of regular expressions without a complement operator as *restricted* regular expressions. Since regular languages are closed under complementation, generalized regular expressions have exactly the same expressive power as restricted regular expressions. However, many languages can be represented more succinctly with generalized regular expressions.

Definition 4.0.4. *(Restricted) regular expressions* over a finite alphabet Σ are defined recursively. $\emptyset$ and a for any $a \in \Sigma$ are regular expressions, and for regular expressions r and s, rs, $r \cup s$ and $r^\star$ are regular expressions. For a regular expression r over Σ, we define the language $\mathcal{L}(r) \subseteq \Sigma^\star$ described by r as follows.

- $\mathcal{L}(\emptyset) := \emptyset$
- $\mathcal{L}(a) := \{a\}$, $a \in \Sigma$
- $\mathcal{L}(rs) := \mathcal{L}(r)\mathcal{L}(s) = \{ww' \mid w \in \mathcal{L}(r), w' \in \mathcal{L}(s)\}$
- $\mathcal{L}(r \cup s) := \mathcal{L}(r) \cup \mathcal{L}(s)$
- $\mathcal{L}(r^\star) := \mathcal{L}(r)^\star = \{w_1 \dots w_n \mid n \in \mathbb{N}, w_1, \dots, w_n \in \mathcal{L}(r)\}$

Definition 4.0.5. *Generalized regular expressions* over a finite alphabet Σ are defined recursively. $\emptyset$ and a for any $a \in \Sigma$ are generalized regular expressions, and for generalized regular expressions r and s, rs, $r \cup s$, $r^\star$ and $\overline{r}$ are generalized regular expressions. The definition of $\mathcal{L}(r)$ for a generalized regular expression r is as for restricted regular expressions, with one additional case.

- $\mathcal{L}(\overline{r}) := \overline{\mathcal{L}(r)} = \Sigma^\star - \mathcal{L}(r)$

Natural measures for the complexity of regular expressions include the size of the expression, the nesting depth of concatenation operators, and the nesting depth of Kleene star operators. The nesting depth of concatenation operators is commonly referred to as *dot-depth*, and the nesting depth of Kleene star operators as *star-height*. Both of these operator nesting measures have received considerable attention in the literature [4, 8, 27, 30, 38, 43, 44]. The main questions to ask here are the following.

- Does an increase in the allowable nesting depth of (generalized) regular expressions lead to an increase in the number of languages that can be described? Is there a limit beyond which a further increase does not allow us to describe more languages?

- Given a language, can we decide what nesting depth is required to describe it? If so, is there an efficient algorithm?

For dot-depth, it has been shown that there is a strict hierarchy [4, 44], meaning that for any $k \in \mathbb{N}^+$ there is a regular language of dot-depth k that cannot be described by any regular expression of dot-depth less than k. This result applies to both restricted and generalized regular expressions. While there has been some progress on the decidability of the dot-depth hierarchy [12, 38], the question is still open. For the star-height of restricted regular expressions, both questions have been answered: The restricted star-height hierarchy is strict [8, 27] and decidable [17, 18]. On the other hand, our knowledge about generalized star-height is extremely limited – we do not even know whether there is a regular language that cannot be expressed without nested stars.

Definition 4.0.6. The *(restricted) star-height* of a regular expression r, $\mathrm{rsh}(r)$, is defined recursively as follows.

- $\mathrm{rsh}(\emptyset) := \mathrm{rsh}(a) = 0$, $a \in \Sigma$
- $\mathrm{rsh}(rs) := \mathrm{rsh}(r \cup s) = \max\{\mathrm{rsh}(r), \mathrm{rsh}(s)\}$
- $\mathrm{rsh}(r^\star) := \mathrm{rsh}(r) + 1$

The *(restricted) star-height* of a regular language $L \subseteq \Sigma^\star$, $\mathrm{rsh}(L)$, is the minimum restricted star-height over all regular expressions that describe L.

$$\mathrm{rsh}(L) := \min\{\mathrm{rsh}(r) \mid \mathcal{L}(r) = L\}$$

Definition 4.0.7. The *generalized star-height* of a generalized regular expression r, $\mathrm{gsh}(r)$, is defined recursively with the same cases as the restricted star-height, where all occurrences of rsh are replaced by gsh, and with one additional case.

- $\mathrm{gsh}(\overline{r}) := \mathrm{gsh}(r)$

The *generalized star-height* of a regular language $L \subseteq \Sigma^\star$, $\mathrm{gsh}(L)$, is the minimum generalized star-height over all generalized regular expressions that describe L.

$$\mathrm{gsh}(L) := \min\{\mathrm{gsh}(r) \mid \mathcal{L}(r) = L\}$$

The languages of generalized star-height 0 are commonly referred to as the *star-free* languages. The following classical theorem gives an exact characterization of the star-free languages.

Theorem 4.0.8 (McNaughton and Papert)**.** [28] Let $L \subseteq \Sigma^+$. Then $L \in \mathrm{FO}$ iff L is regular and $\mathrm{gsh}(L) = 0$.

Are there similar characterizations for the languages of generalized star height k for $k \geq 1$? We do not know. We do not even know whether there is a language of generalized star-height 2.

After reviewing known results on restricted star-height and generalized star height in the following two sections, we investigate the relationship between generalized star-height and first-order logic with transitive closure. The main result of that section is an exact characterization of generalized star-height in terms of first-order logic with a transitive closure operator. While this result does not allow us to solve the generalized star-height problem, we feel that it increases our understanding of generalized star-height, and it makes available a new array of tools from finite model theory to attack the problem, e.g. Ehrenfeucht-Fraïssé games augmented with a transitive closure move. In the remaining three sections of this chapter, we present some of our attempts to construct a language of generalized star height 2.

4.1 Known Results on Restricted Star-Height

The restricted star-height hierarchy is known to be strict [8]. For historic context, we present two alternative proofs here. Both proofs rely on the graph-theoretic notion of cycle rank. The first one is based on constructing hard languages using homomorphisms, while the second one is more graph-theoretic in nature.

For ease of notation, we identify a digraph G with its vertices, and write $v \in G$ when we mean that v is a vertex from G. Additionally, $G - v$ is the subgraph of G induced by the vertex set $G - v$.

Definition 4.1.1. A digraph G is *strongly connected* if for all nodes $u, v \in G$ there is a (possibly empty) path from u to v. A *strongly connected component (SCC)* of G is a maximal strongly connected induced subgraph, and it is called *non-trivial* if it contains at least one edge.

Definition 4.1.2. [8] The *cycle rank* of a digraph G, $\operatorname{crk}(G)$, is inductively defined as

$$\operatorname{crk}(G) := \begin{cases} 0 & \text{if } G \text{ is acyclic} \\ 1 + \min\{\operatorname{crk}(G - v) \mid v \in G\} & \text{if } G \text{ is strongly connected} \\ & \quad \text{and non-trivial} \\ \max\{\operatorname{crk}(C) \mid C \text{ an SCC of } G\} & \text{otherwise} \end{cases}.$$

There is a nice characterization of cycle rank in terms of a cops and robber game [16].

Theorem 4.1.3. [8] If the transition graph of a finite automaton A has cycle rank k, then there is a regular expression e of restricted star-height k such that $\mathcal{L}(A) = \mathcal{L}(e)$.

Proof. We use induction to prove the slightly stronger claim that for any transition graph G of cycle rank k, and any two nodes $x, y \in G$, there is a regular expression of restricted star-height k for all paths from x to y. For $k = 0$, the transition graph G does not contain any cycles, thus the corresponding language is finite and therefore of restricted star-height 0.

For the inductive case, suppose our claim holds for all graphs of cycle rank at most k, and let G be transition graph of cycle rank $k+1$. If G is strongly connected, then we choose $z \in G$ such that $\operatorname{crk}(G - z) = k$. Let $p_1, \ldots, p_s$ be all nodes with outgoing edges to z, and let $q_1, \ldots, q_t$ be all nodes with incoming edges from z. Using the inductive hypothesis, we have a regular expression $\alpha'(x, y)$ of star-height k for all paths from x to y in the graph $G - z$. We write $E_{x,y}$ for the union over all edge labels of directed edges from x to y.

$$\alpha(z, z) = \left(\bigcup_{i=1}^{s} \bigcup_{j=1}^{t} E_{z,q_j} \alpha'(q_j, p_i) E_{p_i,z} \right)^{\star}$$

$$\alpha(x, y) = \alpha'(x, y) \cup \bigcup_{i=1}^{s} \bigcup_{j=1}^{t} \alpha'(x, p_i) E_{p_i,z} \alpha(z, z) E_{z,q_j} \alpha'(q_j, y)$$

If G is not strongly connected, then any path from x to y is of the form

$$c_1 w_1 c_2 w_2 \dots c_{n-1} w_{n-1} c_n$$

where each c_i is a path within an SCC of G, and each w_i is a path from one component to another without intersecting any intermediate components. Since the number of SCCs is finite, the number of direct paths from one component to another is finite, and from the previous argument we have regular expressions of star-height at most $k+1$ for all paths within SCCs. Thus there is a regular expression of star height $k+1$ for all paths in G from x to y. $\square$

Theorem 4.1.4. [8] If a regular language L has restricted star-height k, then there is a non-deterministic finite automaton N whose transition graph has cycle rank at most k such that $L = \mathcal{L}(N)$.

Proof. We use the standard construction of a non-deterministic finite automaton equivalent to a regular expression. Every time a star is encountered in the construction, the cycle rank of the corresponding automaton increases by at most 1. All other operations leave the cycle rank unchanged. $\square$

Corollary 4.1.5 (Eggan)**.** [8] The smallest cycle rank of the transition graphs of all finite automata that recognize a regular language L is equal to $\mathrm{rsh}(L)$.

4.1.1 Eggan's Proof

Definition 4.1.6. The homomorphism $\uparrow_i^k\colon \Sigma_k^\star \to \Sigma_{k+i}^\star$ is defined by $\uparrow_i^k (a_j) := a_{j+i}$. Similarly we define the homomorphism $\downarrow_i^k\colon \Sigma_k^\star \to \Sigma_{k-i+1}^\star$

$$\downarrow_i^k (a_j) := \begin{cases} a_{j-i} & \text{if } j > i \\ a_{k-i+1} & \text{otherwise} \end{cases}.$$

We omit the superscript k when possible.

Definition 4.1.7. Based on the alphabets Σ_k, we define the alphabets

$$\Gamma_k := \Sigma_{2^k-1} = \{a_1, \ldots, a_{2^k-1}\} \quad \text{and} \quad \Lambda_k := \uparrow_{2^k-1} (\Gamma_k) = \{a_{2^k}, \ldots, a_{2^{k+1}-2}\} \quad .$$

We observe that $\Gamma_k = \Gamma_{k-1} \cup \Lambda_{k-1} \cup \{a_{2^k-1}\}$.

Definition 4.1.8. We say that w is a $(1, n)$-word if w contains the subword a_1^n. Moreover, w is a (k, n)-word if it contains a subword of the form $(xuyva_{2^k-1})^n$, where $u \in \Gamma_{k-1}^\star$ is a $(k-1, n)$-word, $v \in \Lambda_{k-1}^\star$ is such that $\downarrow_{2^{k-1}-1} (v)$ is a $(k-1, n)$-word, and x and y are arbitrary.

We note that this definition is slightly more general than necessary – all proofs go through without the arbitrary strings x and y.

Lemma 4.1.9. [8] If L is a regular language such that

(i) L contains (k, n)-words for arbitrarily large n, and

(ii) L does not contain any subwords of the form ab where $a \in \Gamma_i$ and $b \in \Lambda_i$ for $i < k$,

then $\mathrm{rsh}(L) \geq k$.

Proof. We proceed by induction on k. For $k = 1$, we only need to observe that L is infinite and thus $\mathrm{rsh}(L) \geq 1$. Now suppose the lemma holds for k, and let $L \subseteq \Gamma_{k+1}^\star$ such that conditions (i) and (ii) hold, but such that $\mathrm{rsh}(L) \leq k$. Let γ be a regular expression or restricted star-height at most k such that $\mathcal{L}(\gamma) = L$. Since concatenation distributes over union, we may assume that

$$\gamma = \gamma_1 \cup \gamma_2 \cup \ldots \cup \gamma_m \quad \text{where} \quad \gamma_i = x_{i,1}\, \alpha_{i,1}^\star\, x_{i,2}, \alpha_{i,2}^\star, \ldots\, x_{i,s_i}\, \alpha_{i,s_i}^\star\, x_{i,s_i+1}$$

for some $m, s_1, \ldots, s_m \in \mathbb{N}$, $x_{i,j} \in \Gamma_{k+1}^\star$ and $\mathrm{rsh}(\alpha_{i,j}) < k$. We observe that there is some $i \in [1, m]$ such that $\mathcal{L}(\gamma_i)$ already satisfies the conditions of the lemma. Slightly less obvious is the fact that there also is a $j \in [1, s_i]$ such that $\mathcal{L}(\alpha_{i,j}^\star)$ satisfies the conditions of the lemma. Thus $\mathcal{L}(\alpha_{i,j}^\star)$ contains $(k+1, n)$-words for arbitrarily large n.

Now we argue that $L_0 := \mathcal{L}(\alpha_{i,j})$ contains at least one word over Γ_k. Since L_0 has restricted star-height at most $k-1$, it follows from the inductive hypothesis that L_0 cannot contain (k, n)-words for sufficiently large n. Thus L_0^2 does not contain $(k, 2n)$-words. For the sake of a contradiction we assume that L_0 does not contain any words over Γ_k. Then any (k, n)-word in L_0^2 needs to be of the form $xuvy$ with $xu, vy \in L_0$ and $x, y \notin \Gamma_k^\star$ and uv itself being a (k, n)-word. Thus L_0^3 does not contain any $(k, 2n)$-words, and by induction $L_0^\star = \mathcal{L}(\alpha_{i,j}^\star)$ contains no $(k, 2n)$-words, a contradiction. A similar argument shows that L_0 contains at least one word over Λ_k. But this violates condition (ii), a contradiction. □

Theorem 4.1.10. [8] For every k, there is a regular language L_k such that $\mathrm{rsh}(L_k) = k$.

Proof. We inductively define regular expressions to describe the languages L_k. Let

$$\beta_1 = a_1 \quad \text{and} \quad \beta_i = {\beta_{i-1}}^\star \left(\uparrow_{2^{i-1}-1} (\beta_{i-1})\right)^\star a_{2^i-1} \quad .$$

Thus we have (just as examples)

$$\beta_2 = a_1^\star a_2^\star a_3 \quad , \quad \beta_3 = (a_1^\star a_2^\star a_3)^\star (a_4^\star a_5^\star a_6)^\star a_7 \quad .$$

Finally we let $L_k := \mathcal{L}(\beta_k^\star)$. We observe that $\mathrm{rsh}(L_k) \le k$. Since L_k satisfies the conditions of Lemma 4.1.9, we have $\mathrm{rsh}(L_k) = k$. □

4.1.2 McNaughton's Graph-Theoretic Proof

Lemma 4.1.11. [27] If H is an induced subgraph of a digraph G, then $\mathrm{crk}(H) \le \mathrm{crk}(G)$.

Proof. We use induction on the size of G. For $|G| = 1$ the claim is obvious. For $|G| > 1$, let H' be an SCC of H with $\text{crk}(H') = \text{crk}(H)$, and let G' be the SCC of G containing H'. Then we have

$$\begin{aligned}\text{crk}(G) \geq \text{crk}(G') &= 1 + \min\{\text{crk}(G' - v) \mid v \in G'\} \\ &\geq 1 + \min\{\text{crk}(H' - v) \mid v \in H'\} = \text{crk}(H') = \text{crk}(H) \quad .\end{aligned}$$ □

Lemma 4.1.12. [27] Let G be a digraph, and suppose that one of G's SCCs contains two disjoint strongly connected induced subgraphs G_1 and G_2. If $\text{crk}(G_1) \geq k$ and $\text{crk}(G_2) \geq k$, then $\text{crk}(G) \geq k + 1$.

Proof. Let G' be the SCC of G containing both G_1 and G_2. We recall that $\text{crk}(G') = 1 + \min\{\text{crk}(G' - v) \mid v \in G'\}$. Regardless of the choice of $v \in G'$, $G' - v$ contains either G_1 or G_2 as a strongly connected induced subgraph, and thus by the previous lemma $\text{crk}(G' - v) \geq k$. □

Lemma 4.1.13. [27] Let L be a regular language of restricted star-height k, and let G be the transition graph of an automaton accepting L. Then there is a word $w_G \in L$ such that every accepting path in G spelling out w_G intersects an SCC of G of cycle rank at least k.

Proof. Suppose for the sake of a contradiction that there is no such word w_G. Thus the transition graph consisting of only those SCCs of G with cycle rank less than k would still accept L, and therefore $\text{rsh}(L) < k$, a contradiction. □

Theorem 4.1.14. [27] Let Σ_1 and Σ_2 be disjoint alphabets, let $\mathtt{a}$ and $\mathtt{b}$ be letters not in $\Sigma_1 \cup \Sigma_2$, and let $L_1 \subseteq \Sigma_1^\star$ and $L_2 \subseteq \Sigma_2^\star$ be languages of restricted star-height k. The language $L := (\mathtt{a}L_1\mathtt{b}L_2)^\star$ has star height $k + 1$.

Proof. The upper bound is obvious. For the lower bound, let G be the transition graph of a minimal finite automaton that accepts L. To isolate the part of G responsible for L_1, let G_1 be the subgraph of G induced by all nodes with incoming or outgoing edges labeled with letters from Σ_1. We notice that in G, all incoming edges to nodes in G_1 from outside of G_1 must be labeled with $\mathtt{a}$, and all outgoing edges to nodes outside of G_1 must be labeled with $\mathtt{b}$. Thus all of G_1's edges are labeled with letters from Σ_1. Making all nodes of G_1 with incoming $\mathtt{a}$ edges initial nodes, and all nodes of G_1 with outgoing $\mathtt{b}$ edges finial nodes, we get a transition graph for L_1. Thus G_1 has cycle rank at least k. Similarly, we define the subgraph G_2 for L_2, of cycle rank at least k. It is easy to see that G_1 and G_2 are disjoint.

Now let w_1 and w_2 be words for G_1 and G_2, respectively, as stated in Lemma 4.1.13. With $n := \max\{|G_1|, |G_2|\}$, we define the word $w := (\mathtt{a}w_1\mathtt{b}w_2)^{n+1}$. We look at any accepting path of w. Since every portion of this path which corresponds to w_1 must intersect at least one SCC of G_1 of cycle rank at least k, there is at least one such SCC that is intersected twice. Thus we have a path from an SCC of cycle rank at least k in G_1 to an SCC of cycle rank at least k in G_2 and back. By Lemma 4.1.12, G thus has cycle rank at least $k + 1$. □

The construction from the previous theorem uses an alphabet of unbounded size. There is a homomorphism $h : \{a_1, \ldots, a_n\} \to \{0, 1\}$ with $h(a_i) := 10^i10^{n-i+1}1$ which maintains restricted star-height [27]. Thus a binary alphabet is sufficient.

This graph-theoretic approach does not seem to be applicable to the generalized star-height problem. Given languages L_1 and L_2 of generalized star-height $k \geq 1$, the language $L = (\mathtt{a}L_1\mathtt{b}L_2)^\star$ has generalized star-height at most k, since it is the complement of the language

$$\overline{(\mathtt{a}\Sigma_1^\star\mathtt{b}\Sigma_2^\star)^\star} \cup (\Sigma^\star\mathtt{a}(\overline{L_1} \cap \Sigma_1^\star)\mathtt{b}\Sigma^\star) \cup (\Sigma^\star\mathtt{b}(\overline{L_2} \cap \Sigma_2^\star)(\mathtt{a}\Sigma^\star \cup \varepsilon) \quad .$$

4.2 Known Results on Generalized Star-Height

We review some of the main results on the generalized star-height problem from the literature. The underlying motivation to all these results is to identify a language of generalized star-height 2, and to give a characterization of all the languages of generalized star-height 1. The Kleene star operator in regular expressions allows us to do modular counting, as for example in the expression $(\mathtt{aa})^\star$. This counting ability thus distinguishes languages of generalized star-height at least 1 from the star-free languages. Thus it seems natural to look at languages that require a staged or two-level counting process as candidates for languages of generalized star-height at least 2. We first review a historic result by Thomas [43], who presents a family of languages that appear to require two-level counting, but actually do have generalized star-height 1.

We look at languages over the alphabet $\Sigma_2 = \{\mathtt{a}, \mathtt{b}\}$ where all words are concatenations of segments of the form $\mathtt{a}^i\mathtt{b}$ for arbitrary $i \in \mathbb{N}$. This is equivalent to requiring that all words either end with $\mathtt{b}$ or are the empty word. Now we define the languages $W(h, k, r, m)$ where the number of occurrences of segments $\mathtt{a}^i\mathtt{b}$ with $i \equiv r \pmod m$ is congruent to h modulo k.

Theorem 4.2.1 (Thomas)**.** [43] For any $h, k, r, m \in \mathbb{N}$, the language $W(h, k, r, m)$ has generalized star-height at most 1.

A large part of the known results on the generalized star-height problem are based on algebraic language theory. Here we give a brief summary of the main definitions and tools, and refer the reader to the books by Straubing [36] and Pin [29] for more details.

Definition 4.2.2. A *monoid* is a set M together with an associative binary operation $M \times M \to M$ and an identity element $1 \in M$. We typically write $x \cdot y$ or simply xy

for the product of two monoid elements $x, y \in M$, and we also use M to refer to the monoid itself when the operation is understood from the context.

If a monoid contains all inverses with respect to its operation, then it is group.

Definition 4.2.3. A monoid M *recognizes* a language $L \subseteq \Sigma^\star$ if there is a homomorphism $\varphi : \Sigma^\star \to M$ and a set $X \subseteq M$ such that $\varphi^{-1}(X) = L$.

Definition 4.2.4. Let $L \subseteq \Sigma^\star$ be a language. We define the *syntactic congruence* $\sim_L$ of L by saying that $x \sim_L y$ if for all $u, v \in \Sigma^\star$, $uxv \in L$ iff $uyv \in L$. The *syntactic monoid* $M(L)$ of L is the set $L/{\sim_L}$ with the concatenation operation. The *syntactic morphism* of L is the homomorphism $\varphi_L : L \to M(L)$ with $\varphi_L(w) = [w]_L$.

Definition 4.2.5. A monoid M_1 *divides* a monoid M_2, $M_1 \preceq M_2$, if M_1 is the homomorphic image of a submonoid of M_2.

Theorem 4.2.6. A monoid M recognizes a language $L \subseteq \Sigma^\star$ iff $M(L) \preceq M$.

Proof. Suppose that M recognizes L. Then there is a homomorphism $\varphi : \Sigma^\star \to M$ and a set $X \subseteq M$ such that $\varphi^{-1}(X) = L$. We have $\varphi(\Sigma^\star) \leq M$, since for every $x, y \in \varphi(\Sigma^\star)$, there are $u, v \in \Sigma^\star$ such that $\varphi(u) = x$ and $\varphi(v) = y$, and thus $xy = \varphi(u)\varphi(v) = \varphi(uv) \in \varphi(\Sigma^\star)$. We define the map $\psi : \varphi(\Sigma^\star) \to M(L)$ where $\psi(x) = \varphi_L(w)$ for any $w \in \Sigma^\star$ with $\varphi(w) = x$. To see that ψ is well-defined, suppose that $x = \varphi(w) = \varphi(w')$. Thus for all $u, v \in \Sigma^\star$, $\varphi(uwv) = \varphi(u)\varphi(w)\varphi(v) = \varphi(uw'v)$, and $uwv \in L$ iff $uw'v \in L$. Therefore $w \sim_L w'$ and $\varphi_L(w) = \varphi_L(w')$. To see that ψ is a homomorphism, let $x = \varphi(u)$ and $y = \varphi(v)$. Since φ is a homomorphism, we have $xy = \varphi(u)\varphi(v)$, and thus $\psi(xy) = \varphi_L(uv) = \varphi_L(u)\varphi_L(v) = \psi(x)\psi(y)$. Finally, we observe that ψ is surjective, since for every $x \in M(L)$, there is a $w \in \Sigma^\star$ with $\varphi_L(w) = x$ and thus $\psi(\varphi(w)) = x$.

For the other direction suppose that $M(L) \preceq M$. Thus there are a submonoid $M' \leq M$ and a surjective homomorphism $\psi : M' \to M(L)$. We define the homomorphism $\varphi : \Sigma^\star \to M'$ by setting $\varphi(\varepsilon)$ to the identity of M', for every $a \in \Sigma$, $\varphi(a) = x$

$$\begin{tikzcd}
 & \Sigma^\star \arrow[dl, "\varphi_L"'] \arrow[d, "\varphi"] \\
M(L) & \varphi(\Sigma^\star) \leq M \arrow[l, "\psi"]
\end{tikzcd}$$

Figure 4.1. Homomorphisms in the proof of Theorem 4.2.6.

for some fixed $x \in M'$ where $\psi(x) = \varphi_L(a)$, and inductively for $w \in \Sigma^\star$, $\varphi(wa) = \varphi(w)\varphi(a)$. We let $X = \psi^{-1}(\varphi_L(L))$. For every $w \in \Sigma^\star$ we have $\psi(\varphi(w)) = \varphi_L(w)$, and thus

$$\varphi(w) \in X \quad \text{iff} \quad \psi(\varphi(w)) \in \varphi_L(L) \quad \text{iff} \quad \varphi_L(w) \in \varphi_L(L) \quad \text{iff} \quad w \in L \quad . \qquad \square$$

Several classes of monoids (more precisely, pseudo-varieties of monoids) have been identified to be of particular importance, and to correspond to well-known classes of languages.

Theorem 4.2.7 (Myhill, Nerode)**.** A language is regular iff its syntactic monoid is finite.

Theorem 4.2.8 (Schützenberger)**.** [33] A language is star-free iff its syntactic monoid is aperiodic, i.e. there is no nontrivial group that divides it.

Definition 4.2.9. A group G is *nilpotent* of class $n \in \mathbb{N}^+$ if the sequence $G_1 = G$, $G_{i+1} = [G_i, G]$ reaches $G_{n+1} = \{1\}$ and $G_n \neq \{1\}$. Here $[G, H] = \{g^{-1}h^{-1}gh \mid g \in G, h \in H\}$ is the commutator group of G and H.

We contrast this definition with the notion of solvability for groups.

Definition 4.2.10. A group G is *solvable* if the sequence $G_1 = G$, $G_{i+1} = [G_i, G_i]$ reaches $G_n = \{1\}$ for some $n \in \mathbb{N}^+$.

Thus every nilpotent group is solvable, but solvable groups need not be nilpotent.

Theorem 4.2.11 (Thérien)**.** [41] The syntactic monoid of a language $L \subseteq \Sigma^\star$ is a nilpotent group of class n iff L it is a boolean combination of languages that count the number of occurrences of subsequences of length at most n modulo some integer.

Theorem 4.2.12 (Pin, Straubing, Thérien)**.** [30] If the syntactic monoid of a language is nilpotent of class 2, then it has generalized star-height at most 1.

Pin, Straubing and Thérien [30] prove an even stronger version of this theorem, showing that any language that is a boolean combination of languages that count occurrences of subsequences of length at most 3 modulo a square-free integer has generalized star-height at most 1.

While a common theme of previous theorems is to understand generalized star height in terms of syntactic monoids, the following theorem suggests that the syntactic monoid alone is unlikely to help with the classification of languages of restricted star-height greater than 1. In particular, it says that any regular language is recognized by a monoid that is the syntactic monoid of a language of restricted star-height at most 1. Even so, we still believe that syntactic monoids might play a crucial role in solving the generalized star-height problem.

Theorem 4.2.13. [30] For every regular language $L \subseteq \Sigma^\star$, there is a language $K \subseteq \Gamma^\star$ and a homomorphism $\varphi : \Sigma^\star \to \Gamma^\star$ such that K has restricted star-height at most 1 and $L = \varphi^{-1}(K)$.

Proof. Let $D = (Q, \Sigma, \delta, q_0, F)$ with $Q = \{q_0, \ldots, q_{n-1}\}$ be a DFA that accepts L, and let $\Gamma = \Sigma \cup \{\#\}$ where $\# \notin \Sigma$. We define the following regular languages.

$$P := \{\#^i a \#^{n-\delta(q_i,a)} \mid 0 \leq i < n, a \in \Sigma\}$$

$$S := \{\#^i \mid q_i \in F\}$$

$$K := P^\star S$$

We observe that both P and S are finite, and thus K has restricted star-height at most 1. We define the homomorphism $\varphi : \Sigma^\star \to \Gamma^\star$ with $\varphi(a) = a\#^n$ for $a \in \Sigma$, and see that $L = \varphi^{-1}(K)$. $\square$

4.3 First-Order Logic with Transitive Closure

There is a natural correspondence between regular expressions and first-order logic with a transitive closure operator. This connection allows us to use Ehrenfeucht-Fraïssé games, extended with a transitive closure move, to prove that some regular languages cannot be described with regular expressions of a given complexity.

Definition 4.3.1. [19, 21] FO[TC] is the extension of FO with a monadic reflexive transitive closure operator. FO[TC] contains all formulas $(\mathrm{TC}_{u,v}\ \varphi)(x,y)$ where $\varphi \in$ FO[TC], and is closed under all first-order operators. The transitive closure operator TC binds the free occurrences of u and v in φ. FO[TC_k] is the subset of FO[TC] of formulas with at most k nested transitive closure operators.

To define the semantics of the transitive closure operator on finite words, let w be a finite word structure that interprets the free variables of $(\mathrm{TC}_{u,v}\ \varphi)(x,y)$. Then $w \models (\mathrm{TC}_{u,v}\ \varphi)(x,y)$ iff there are $n \in \mathbb{N}$ and $\ell_0,\ldots,\ell_n \in |w|$ where $\ell_0 = x^w$ and $\ell_n = y^w$ such that for all $i \in [0,n)$, $(w, \ell_i/u, \ell_{i+1}/v) \models \varphi$. We call the positions $\ell_0,\ldots,\ell_n$ the *path points* of this transitive closure application.

Since TC is a reflexive transitive closure operator, for any word w, any formula φ, and any $i \in |w|$, we have $(w,i,i) \models (\mathrm{TC}_{u,v}\ \varphi)(x,y)$. We also note that the two variables u and v do not necessarily have to be distinct from x and y, but we usually choose different variable names for the sake of clarity.

Using this transitive closure operator, we can translate generalized regular expressions into first-order formulas, where the generalized star-height of the regular expression corresponds exactly to the nesting depth of transitive closure operators in the first-order formula. One technical complication here is that there is no logical structure corresponding to the empty string ε.

Lemma 4.3.2. Let e be a generalized regular expression of generalized star height at most k. Then there is a formula $\varphi(x,y) \in \mathrm{FO}[\mathrm{TC}_k]$ such that for all $w \in \Sigma^+$, and for all $\ell, m \in [1, \|w\|]$ with $\ell \leq m$, $w_{[\ell,m)} \in \mathcal{L}(e)$ iff $(w, \ell, m) \models \varphi(x,y)$.

Proof. We use structural induction on e. For the inductive cases, let $\psi(x,y)$ and $\xi(x,y)$ be the formulas corresponding to the expressions f and g, respectively.

- If $e = \emptyset$, then we choose $\varphi = \bot$.

- If $e = a$ for some $a \in \Sigma$, then we choose $\varphi(x,y) = Q_a(x) \wedge \mathrm{Suc}(x,y)$. Suppose that $w_{[\ell,m)} \in \mathcal{L}(e)$. Then $m = \ell + 1$ and $w_\ell = a$, thus $(w, \ell, m) \models \varphi(x,y)$. On the other hand, if $(w, \ell, m) \models \varphi(x,y)$, then $m = \ell + 1$ and thus $w_{[\ell,m)} = w_\ell = a$.

- If $e = f \cup g$, then we choose $\varphi(x,y) = \psi(x,y) \vee \xi(x,y)$.

- If $e = \overline{g}$, then $\varphi(x,y) = \neg\psi(x,y)$.

- If $e = fg$, we choose

  $$\varphi = \exists z \, (x \leq z \leq y \wedge \psi(x,z) \wedge \xi(z,y)) \quad .$$

  Suppose that $w_{[\ell,m)} \in \mathcal{L}(e)$. Then there is a $p \in [\ell, m]$ such that $w_{[\ell,p)} \in \mathcal{L}(f)$ and $w_{[p,m)} \in \mathcal{L}(g)$. Using the inductive hypothesis, we have both $(w, \ell, p) \models \psi(x,y)$ and $(w, p, m) \models \xi(x,y)$. Thus

  $$(w, \ell, m, p) \models x \leq z \leq y \wedge \psi(x,z) \wedge \xi(z,y)$$

  and $(w, \ell, m) \models \varphi$.

For the other direction, suppose that $(w, \ell, m) \models \varphi$. Then there is a $p \in [\ell, m]$ such that $(w, \ell, p) \models \psi(x, y)$ and $(w, p, m) \models \xi(x, y)$. By the inductive hypothesis, we have $w_{[\ell,p)} \in \mathcal{L}(f)$ and $w_{[p,m)} \in \mathcal{L}(g)$, hence $w_{[\ell,m)} \in \mathcal{L}(e)$.

- If $e = f^\star$, then we choose $\varphi(x, y) = (\mathrm{TC}_{u,v}\ u < v \wedge \psi(u, v))(x, y)$. Suppose that $w_{[\ell,m)} \in \mathcal{L}(e)$. Then there are $n \in \mathbb{N}$ and $p_0, \ldots, p_n$ such that $p_0 = \ell$, $p_n = m$, $p_0 < \ldots < p_n$, and for all $i \in [0, n)$, $w_{[p_i,p_{i+1})} \in \mathcal{L}(f)$. Using the inductive hypothesis, we have $(w, p_i, p_{i+1}) \models \psi(x, y)$ for all $i \in [0, n)$, and thus $(w, \ell, m) \models \varphi(x, y)$.

  For the other direction, suppose that $(w, \ell, m) \models \varphi(x, y)$. Then there are $n \in \mathbb{N}$ and $p_0, \ldots, p_n \in |w|$ such that $p_0 = \ell$, $p_n = m$, and for all $i \in [0, n)$, $p_i < p_{i+1}$ and $(w, p_i, p_{i+1}) \models \psi(x, y)$. Using the inductive hypothesis, we have $w_{[p_i,p_{i+1})} \in \mathcal{L}(f)$ for all $i \in [0, n)$, and thus $w_{[\ell,m)} \in \mathcal{L}(e)$. □

This lemma only applies to substrings of finite words that do not include the last letter of the word, and not to full finite words. To formulate the lemma for finite words, one solution would be to modify our notion of word structure so that every word includes an extra marker at the end. Because we want to stay more in line with the other results of this chapter, in particular with Potthoff's Theorem (Theorem 4.3.5), we instead choose to isolate the last letter and handle it separately.

Lemma 4.3.3. Let $L \subseteq \Sigma^+$ be a regular language of generalized star height at most k. Then there are regular expressions e_a of generalized star height at most k for every $a \in \Sigma$ such that

$$L = \mathcal{L}\left(\bigcup_{a \in \Sigma} e_a a\right) \quad .$$

Proof. Let e be a generalized regular expression of generalized star-height at most k such that $\mathcal{L}(e) = L$. We construct the expressions e_a by induction on the structure of e.

- If $e = \emptyset$, $e = \varepsilon$, or $e \in \Sigma - \{a\}$, then $e_a = \emptyset$.
- If $e = a$, then $e_a = \varepsilon$.
- If $e = f \cup g$, then $e_a = f_a \cup g_a$.
- If $e = \overline{f}$, then $e_a = \overline{f_a}$.
- If $e = fg$, we need to consider whether $\varepsilon \in \mathcal{L}(g)$. If this is the case, then $e_a = fg_a \cup f_a$, otherwise $e_a = fg_a$.
- If $e = f^\star$, then $e_a = f^\star f_a$.

We claim that $\mathcal{L}(e_a) = \{w \mid wa \in \mathcal{L}(e)\}$. The base cases and the case for unions obviously hold. In the case for complementation, we have

$$\mathcal{L}(\overline{f_a}) = \overline{\mathcal{L}(f_a)} = \overline{\{w \mid wa \in \mathcal{L}(f)\}} = \{w \mid wa \notin \mathcal{L}(f)\} = \{w \mid wa \in \mathcal{L}(\overline{f})\} \quad .$$

For concatenation, we have

$$\mathcal{L}(fg_a) = \{ww' \mid w \in \mathcal{L}(f), w' \in \mathcal{L}(g_a)\} = \{ww' \mid w \in \mathcal{L}(f), w'a \in \mathcal{L}(g)\} \quad .$$

If $\varepsilon \notin \mathcal{L}(g)$, then

$$\mathcal{L}(fg_a) = \{ww' \mid w \in \mathcal{L}(f), w'a \in \mathcal{L}(g)\} = \{w \mid wa \in \mathcal{L}(fg)\} \quad .$$

Otherwise

$$\begin{aligned}\mathcal{L}(fg_a \cup f_a) &= \{ww' \mid w \in \mathcal{L}(f), w'a \in \mathcal{L}(g)\}\} \cup \{w \mid wa \in \mathcal{L}(f)\} \\ &= \{w \mid wa \in \mathcal{L}(fg)\} \quad .\end{aligned}$$

Finally, for Kleene star, we have

$$\begin{aligned}\mathcal{L}(f^\star f_a) = \{ww' \mid w \in \mathcal{L}(f^\star), w' \in \mathcal{L}(f_a)\} &= \{ww' \mid w \in \mathcal{L}(f^\star), w'a \in \mathcal{L}(f)\} \\ &= \{w \mid wa \in \mathcal{L}(f^\star f)\} \\ &= \{w \mid wa \in \mathcal{L}(f^\star)\} \quad .\end{aligned}$$

The last equality holds because $wa \neq \varepsilon$. $\square$

Lemma 4.3.4. Let $L \subseteq \Sigma^+$ be a regular language with $\text{gsh}(L) = k$. Then $L \in \text{FO}[\text{TC}_k]$.

Proof. Let e be a generalized regular expression of generalized star-height at most k such that $\mathcal{L}(e) = L$. We apply Lemma 4.3.3 to convert e into an equivalent generalized regular expression of the same generalized star-height of the form $\bigcup_{a\in\Sigma} e_a a$. Using Lemma 4.3.2, we find formulas $\varphi_a(x, y)$ for every $a \in \Sigma$ such that for all $w \in \Sigma^+$, $w_{[1,\|w\|)} \in \mathcal{L}(e_a)$ iff $(w, 1, \|w\|) \models \varphi_a(x, y)$. We define the $\text{FO}[\text{TC}_k]$ sentence

$$\varphi := \bigvee_{a\in\Sigma} (\varphi_a(\min, \max) \wedge Q_a(\max)) \quad ,$$

and argue that $L = \mathcal{L}(\varphi)$. For one direction, suppose that $w \in L$. Let $a = w_{\|w\|}$ be the last letter of w. Thus $w_{[1,\|w\|)} \in \mathcal{L}(e_a)$. Hence we have $(w, 1, \|w\|) \models \varphi_a(x, y)$, thus $w \models \varphi_a(\min, \max)$ and $w \models Q_a(\max)$. Therefore $w \models \varphi$. For the other direction, suppose that $w \models \varphi$. Then there is an $a \in \Sigma$ such that $w \models \varphi_a(\min, \max)$ and $w \models Q_a(\max)$. Hence $(w, 1, \|w\|) \models \varphi_a(x, y)$, thus $w_{[1,\|w\|)} \in \mathcal{L}(e_a)$ and $w_{\|w\|} = a$. Therefore $w \in \mathcal{L}(e_a a)$ and $w \in L$. $\square$

This theorem allows us to translate generalized regular expressions into FO[TC] formulas while maintaining a correspondence between generalized star height and the nesting depth of transitive closure operators. The following theorem shows that this translation is possible with transitive closure nesting depth of only 2, and thus the logic FO[TC] does not give us an exact characterization of generalized star height, unless the generalized star height hierarchy collapses to the second level.

Theorem 4.3.5 (Potthoff)**.** [32] Let $L \subseteq \Sigma^+$ be a regular language. Then $L \in \mathrm{FO}[\mathrm{TC}_2]$.

Proof. Let $A = (\Sigma, Q, \delta, q_0, F)$ be a deterministic finite automaton such that $\mathcal{L}(A) = L$. To simplify notation, we assume that $q_0 = 0$ and $Q = [0, |Q|)$. We write a formula that processes the input in chunks of $|Q|$ symbols at a time. To find the initial position of each chunk, we define the formula

$$\gamma(x) := \left(\mathrm{TC}_{u,v}\ \mathrm{DIST}_{|Q|}(u,v)\right)(\min, x) \quad ,$$

where the formula $\mathrm{DIST}_i(x,y)$ says that y is exactly i positions to the right of x. Thus $(w,i) \models \gamma(x)$ iff $i \equiv 1 \pmod{|Q|}$. To process one chunk of $|Q|$ symbols, we define the formula

$$\varphi_0(x,y) := \bigvee_{q,q' \in Q, s \in \Sigma^{|Q|} : \delta^\star(q,s) = q'} \exists u\, (\gamma(u) \wedge \mathrm{DIST}_q(u,x) \wedge \exists v\, (\gamma(v) \wedge \mathrm{DIST}_{q'}(v,y) \\ \wedge \mathrm{READ}_s(u,v))) \quad ,$$

where $(w,i,j) \models \mathrm{READ}_s(x,y)$ iff $w_{[i,j)} = s$. The formulas DIST and READ are easily constructed using existential quantifiers and the successor relation.

We use a transitive closure operator to process all full chunks of $|Q|$ letters, and handle the remaining symbols separately. Let $k \in [0, |Q|)$ such that $\|w\| \equiv k \pmod{|Q|}$, and let

$$\varphi := \bigvee_{s \in L : \|s\| < |Q|} \mathrm{READ}'_s(\min, \max) \vee \exists x\ ((\mathrm{TC}_{u,v}\ \varphi_0(u,v))(\min, x) \wedge \varphi_1(x)) \quad ,$$

where

$$\varphi_1(x) := \bigvee_{q\in Q, s\in\Sigma^{|Q|+k}:\delta^\star(q,s)\in F} \exists u\,(\gamma(u) \wedge \mathrm{DIST}_q(u,x) \wedge \mathrm{READ}'_s(u,\max))$$

and $\mathrm{READ}'_s(x,y)$ is just like $\mathrm{READ}_s(x,y)$, except that it also reads the letter at the position of y, i.e. $(w,i,j) \models \mathrm{READ}'_s(x,y)$ iff $w_{[i,j]} = s$.

We claim that $L = \mathcal{L}(\varphi)$. For the first direction, suppose that $w \in L$. If $\|w\| < |Q|$, then $w \models \mathrm{READ}'_w(\min,\max)$ and thus $w \models \varphi$. Otherwise let $\ell \in \mathbb{N}$ and $k \in [0,|Q|)$ such that $\|w\| = (\ell+1)\cdot|Q| + k$. For all $i \in [0,\ell]$, let q_i be the state of the automaton after reading the first $i\cdot|Q|$ letters of w. Thus, for all $i \in [0,\ell)$, we have $\delta^\star(q_i, w_{(i\cdot|Q|,(i+1)\cdot|Q|]}) = q_{i+1}$, and therefore

$$(w, i\cdot|Q|+1+q_i, (i+1)\cdot|Q|+1+q_{i+1}) \models \varphi_0(x,y) \quad .$$

Hence $(w, 1, \ell\cdot|Q|+1+q_\ell) \models (\mathrm{TC}_{u,v}\ \varphi_0(u,v))\,(x,y)$. Additionally, $\delta^\star(q_\ell, w_{(\ell\cdot|Q|,\|w\|]}) \in F$, and thus

$$(w, \ell\cdot|Q|+1+q_\ell, \|w\|) \models \varphi_1(x) \quad .$$

Therefore $w \models \varphi$.

For the other direction, suppose that $w \models \varphi$. If $w \models \mathrm{READ}'_s(\min,\max)$ for some $s \in L$ with $\|s\| < |Q|$, then $w = s$ and $w \in L$. Otherwise there is an $j \in \|w\|$ such that

$$(w,j) \models (\mathrm{TC}_{u,v}\ \varphi_0(u,v))\,(\min,x) \wedge \varphi_1(x) \quad .$$

Thus there are $\ell \in \mathbb{N}$ and $p_0,\ldots,p_\ell \in \|w\|$ such that $p_0 = 1$, $p_\ell = j$, $p_0 < \ldots < p_\ell$, and for all $i \in [0,\ell)$, $(w,p_i,p_{i+1}) \models \varphi_0(x,y)$. Hence there are $q_i, q'_i \in Q$, $s_i \in \Sigma^{|Q|}$ with $\delta(q_i,s_i) = q'_i$, and $u_i, v_i \in \|w\|$ such that

$$(w, p_i, p_{i+1}, u_i/u, v_i/v) \models \gamma(u) \wedge \mathrm{DIST}_q(u,x) \wedge \gamma(v) \wedge \mathrm{DIST}_{q'}(v,y) \wedge \mathrm{READ}_s(u,v) \quad .$$

Thus $u_i, v_i \equiv 1 \pmod{|Q|}$, $v_i - u_i = |Q|$, and $w_{[u_i,v_i)} = s_i$. Therefore the automaton transitions from state q_i to state q_i' reading $w_{[u_i,v_i)}$. Because u_i and v_i are uniquely determined by their distance to p_i and p_{i+1}, respectively, we have $q_{i+1} = q_i'$ and $u_{i+1} = v_i$. Therefore

$$\delta^\star(q_0, w_{[1,v_{\ell-1})}) = \delta^\star(\ldots\delta^\star(q_0, w_{[u_0,v_0)}), \ldots, w_{[u_{\ell-1},v_{\ell-1})}) = q_{\ell-1}' = q_\ell \quad .$$

For the final step, since $(w, j) \models \varphi_i(x)$, there are $q \in Q$ and $s \in \Sigma^{|Q|+k}$ with $\delta^\star(q, s) \in F$, and there is a $t \in \|w\|$ such that

$$(w, j, t/u) \models \gamma(u) \wedge \mathrm{DIST}_q(u, x) \wedge \mathrm{READ}_s'(u, \max) \quad .$$

Hence $t = v_{\ell-1}$, $q = q_\ell$ and $s = w_{[v_{\ell-1},\|w\|]}$, therefore $\delta^\star(q_0, w) \in F$, and $w \in \mathcal{L}(A) = L$. $\square$

Corollary 4.3.6. If every regular language from FO[TC_2] has generalized star-height at most k, then every regular language has generalized star-height at most k, i.e. the generalized star-height hierarchy collapses to level k.

Open Problem 4.3.7. What can we say about the generalized star-height of the languages from FO[TC_1] and FO[TC_2]? Is there a language in FO[TC_2] – FO[TC_1]?

Building on another classical result, we at least know that every FO[TC_k] formula describes a regular language, and thus corresponds to some generalized regular expression.

Theorem 4.3.8 (Büchi, Elgot, Trakhtenbrot)**.** [5,6,45] Let $L \subseteq \Sigma^\star$. Then $L \in$ MSO iff L is regular.

Theorem 4.3.9. FO[TC] on finite words captures exactly the regular languages.

Proof. We first observe that the transitive closure operator can be easily translated into monadic second order logic.

$$\begin{aligned}
&w \models (\mathrm{TC}_{x,y}\ \varphi)(x,y) \\
&\qquad\qquad\text{iff}\quad w \models \forall X\,((X(x) \wedge \forall x \forall y\,((X(x) \wedge \varphi(x,y)) \to X(y))) \to X(y))
\end{aligned}$$

Thus every FO[TC] formula is equivalent to a monadic second order formula. Using Theorem 4.3.8, this implies the formula describes a regular language. The other direction follows from both Lemma 4.3.4 and Theorem 4.3.5. □

4.3.1 Regular Formulas and Forward Transitive Closure

To provide an exact correspondence between transitive closure depth and generalized star height, we consider a restricted version of FO[TC].

Definition 4.3.10. We define the *forward transitive closure* operator FTC by

$$(\mathrm{FTC}_{u,v}\ \varphi)(x,y) := (\mathrm{TC}_{u,v}\ u < v \wedge \varphi)(x,y) \quad .$$

As usual, we write FO[FTC] to refer to first-order logic with FTC, and FO[FTC_k] for FO[FTC] formulas of transitive closure nesting depth at most k.

Theorem 4.3.11 (Potthoff)**.** [32] Let $L \subseteq \Sigma^+$ be a regular language. Then $L \in$ FO[FTC_2].

Proof. We observe that all TC operators in the proof of Theorem 4.3.5 are FTC operators. □

Definition 4.3.12. *Regular* first-order logic with FTC, rFO[FTC], is the closure under boolean combinations of all formulas $\varphi(x,y)$ of the following forms.

- $x = y$, $x < y$, and $\mathrm{Suc}(x,y)$

- $Q_a(x) \wedge x < y$
- $\exists z\,(x \leq z \leq y \wedge \psi(x,z) \wedge \xi(z,y))$, where $\psi, \xi \in$ rFO[FTC]
- $(\mathrm{FTC}_{u,v}\; u < v \wedge \psi(u,v))\,(x,y)$, where $\psi \in$ rFO[FTC]

Intuitively, any rFO[FTC] formula with exactly two free variables x and y is interpreted only over the positions in between x and y. The letter at position y, and all letters to the left of x or to the right of y are irrelevant to this formula.

Every formula in rFO[FTC] has at least two free variables, and only if it is a boolean combination can it have more than two free variables. Whenever we refer to a formula as $\varphi(x,y)$, this notation implies that x and y are the only free variables of φ.

Lemma 4.3.13. Let $\varphi(x,y) \in$ rFO[FTC$_k$]. Then there is a generalized regular expression e such that $\mathrm{gsh}(e) = k$ and for all $w \in \Sigma^+$, and all $m, n \in |w|$ with $m \leq n$, we have $w_m \ldots w_{n-1} \in \mathcal{L}(e)$ iff $(w,m,n) \models \varphi(x,y)$.

Proof. We construct e by induction on the structure of φ. For $\varphi = x = y$, $e = \varepsilon$. For $\varphi = x < y$, $e = \Sigma^+$. For $\varphi = \mathrm{Suc}(x,y)$, $e = \Sigma$. For $\varphi = Q_a(x) \wedge x < y$, we choose $e = a\Sigma^\star$. The cases for boolean combinations are trivial.

In the case where $\varphi(x,y) = \exists z\,(x \leq z \leq y \wedge \psi(x,z) \wedge \xi(z,y))$, we apply the inductive hypothesis to both $\psi(x,z)$ and $\xi(z,y)$, and have corresponding generalized regular expressions f and g. We choose $e = fg$. In the case where $\varphi(x,y) = (\mathrm{FTC}_{u,v}\; u < v \wedge \psi(u,v))\,(x,y)$, we set $e = f^\star$, where f is the generalized regular expression corresponding to $\psi(u,v)$. $\square$

For an exact correspondence between regular FTC formulas and generalized regular expressions, we extend rFO[FTC] to allow us to describe the letter at the last position of a word.

Definition 4.3.14. The logic $\mathrm{rFO}^+[\mathrm{FTC}]$ is the boolean closure of all sentences $\varphi(\min, \max)$ where $\varphi(x, y) \in \mathrm{rFO}[\mathrm{FTC}]$, and the sentences $Q_a(\max)$ where $a \in \Sigma$.

Theorem 4.3.15. Let $L \subseteq \Sigma^+$ be a regular language. Then $\mathrm{gsh}(L) = k$ iff $L \in \mathrm{rFO}^+[\mathrm{FTC}_k]$.

Proof. The only TC operators in the proof of Lemma 4.3.4 are applied to formulas $\varphi(u, v)$ that imply $u < v$. Thus they are FTC operators, and the formula constructed in that lemma is in $\mathrm{rFO}^+[\mathrm{FTC}_k]$.

For the other direction, if $\varphi = \psi(\min, \max)$ for some $\psi(x, y) \in \mathrm{rFO}[\mathrm{FTC}_k]$, then let e be the corresponding generalized regular expression as in 4.3.13. Thus for all $w \in \Sigma^+$, $w_1 \ldots w_{\|w\|-1} \in \mathcal{L}(e)$ iff $(w, 1, \|w\|) \models \varphi(x, y)$. The latter condition is equivalent to $w \models \varphi(\min, \max)$. Thus we satisfy all requirements with the expression $e\Sigma$.

If $\varphi = Q_a(\max)$ for some $a \in \Sigma$, we simply choose the expression $\Sigma^\star a$. Finally, for boolean combinations we only need to apply the inductive hypothesis to the parts of the boolean combination. $\square$

Using the last theorem, we are now able to express the generalized star height problem in terms of first-order logic with forward transitive closure.

Open Problem 4.3.16. Is there a $k \in \mathbb{N}$ such that $\mathrm{rFO}^+[\mathrm{FTC}_{k+1}] = \mathrm{rFO}^+[\mathrm{FTC}_k]$, and thus $\mathrm{FO}[\mathrm{TC}_2] \subseteq \mathrm{rFO}^+[\mathrm{FTC}_k]$?

Open Problem 4.3.17. Is there a language in $\mathrm{rFO}^+[\mathrm{FTC}_2] - \mathrm{rFO}^+[\mathrm{FTC}_1]$?

4.3.1.1 Game Characterizations of FO[FTC] and rFO$^+$[FTC]

We define two variations on the classical Ehrenfeucht-Fraïssé game that characterize $\mathrm{FO}[\mathrm{FTC}_k]$ and $\mathrm{rFO}^+[\mathrm{FTC}_k]$. Similar games for logics with an unrestricted TC operator were introduced in [13] and [7].

Definition 4.3.18. Let $n, k \in \mathbb{N}$, and let $w, w' \in \Sigma^\star$. The $\mathrm{FO}_n[\mathrm{FTC}_k](w, w')$ game is the extension of the $\mathrm{FO}_n(w, w')$ game with the FTC move. In this move, Samson picks one of the two structures, say w, and he chooses two pebbles that are already placed on the board, call them x and y. He then temporarily marks an arbitrary number of positions in between x and y on w, and he also marks the positions with the pebbles x and y on both structures. Delilah replies by marking a (possibly different) number of positions in between x and y on w'. Now Samson again picks two pebbles r and s (not necessarily different from x and y), and places them on two consecutive marked positions on w'. Delilah replies by putting her corresponding pebbles on two consecutive marked positions on w. At the end of the move, all marks are cleared.

The game $\mathrm{FO}_n[\mathrm{FTC}_k](w, w')$ has at most n existential moves and at most k FTC moves.

Definition 4.3.19. Let Var be the set of all variables. A *game configuration* on the pair of structures (w, w') is determined by the positions of the pebbles on the board, i.e. by two partial maps, $\alpha : \mathrm{Var} \to |w|$ and $\beta : \mathrm{Var} \to |w'|$. For $\ell \in \mathbb{N}$, an *ℓ-configuration* is a pair of maps that assign positions to exactly ℓ variables.

Theorem 4.3.20. Let w and w' be finite words, and let $k, n \in \mathbb{N}$. Delilah has a winning strategy in the game $\mathrm{FO}_n[\mathrm{FTC}_k](w, w')$ iff w and w' agree on all $\mathrm{FO}_n[\mathrm{FTC}_k]$ sentences.

Proof. We use induction to prove the stronger statement that also captures configurations during the course of the game. Every existential move introduces at most one new pebble, and every FTC move introduces at most two new pebbles. In correspondence to this, every existential quantifier introduces at most one new variable, and every FTC operator introduces at most two new variables.

We claim that for all $(n+2k)$-configurations (α, β), Delilah has a winning strategy for the game $\mathrm{FO}_n[\mathrm{FTC}_k](w, w')$ in configuration (α, β) iff (w, α) and (w', β) agree on all $\mathrm{FO}_n[\mathrm{FTC}_k]$ formulas whose free variables are a subset of the domains of α and β.

Suppose that there is a formula $\varphi \in \mathrm{FO}_n[\mathrm{FTC}_k]$ with all free variables from the domains of α and β such that $(w, \alpha) \models \varphi$ and $(w', \beta) \not\models \varphi$. If φ is atomic, then Delilah just lost the game. If φ is $\neg\psi$ then apply our claim with the roles of w and w' reversed. If $\varphi = \psi \vee \xi$, then we can apply our claim to one of ψ or ξ. If $\varphi = \exists x\, \psi$, we have Samson place his pebble x on a position $i \in |w|$ such that $(w, \alpha, i/x) \models \psi$. Regardless of Delilah's reply $j \in |w'|$, we have $(w', \beta, j/x) \not\models \psi$. Applying the inductive hypothesis, we see that Delilah does not have a winning strategy for the remaining game.

Finally, if $\varphi = (\mathrm{FTC}_{p,q}; \psi)\,(x, y)$, then we let Samson mark a valid sequence of path points in between x and y. Because $(w', \beta) \not\models \varphi$, no matter how Delilah replies, there are two consecutive marked positions $j_1, j_2 \in |w'|$ such that $(w', \beta, j_1/p, j_2/q) \not\models \psi$. Samson places p and q on those two positions, and no matter which two consecutive marked positions in $i_1, i_2 \in |w|$ Delilah chooses to reply with, we have $(w, \alpha, i_1/p, i_2/q) \models \psi$. Thus we can apply the inductive hypothesis, and see that Delilah does not have a winning strategy for the remaining game.

For the other direction, suppose that (w, α) and (w', β) agree on all $\mathrm{FO}_n[\mathrm{FTC}_k]$ formulas whose free variables are a subset of the domains of α and β. If Samson's first move in the game on (w, w') in configuration (α, β) is an existential move, placing pebble x on position $i \in |w|$, then we let Φ be the conjunction of all finitely many inequivalent $\mathrm{FO}_{n-1}[\mathrm{FTC}_k]$ formulas satisfied by $(w, \alpha, i/x)$. Thus $(w, \alpha) \models \exists x\, \Phi$, and by our assumption $(w', \beta) \models \exists x\, \Phi$. We have Delilah place her pebble x at $j \in |w'|$ such that $(w', \beta, j/x) \models \Phi$. Since Φ is a complete description of all the formulas satisfied by $(w, \alpha, i/x)$, the inductive hypothesis applies and Delilah has a winning strategy for the remainder of the game.

Otherwise Samson's first move is an FTC move. Suppose he marks positions $\ell_0, \ldots, \ell_r \in |w|$. For every $i \in [0, r)$, let Φ_i be the conjunction of all finitely many inequivalent $\mathrm{FO}_n[\mathrm{FTC}_{k-1}]$ formulas satisfied by $(w, \alpha, \ell_i/p, \ell_{i+1}/q)$. Thus $(w, \alpha) \models \left(\mathrm{FTC}_{p,q}\ \bigvee_{i \in [0,r)} \Phi_i\right)(x, y)$, and by assumption $(w', \beta) \models \left(\mathrm{FTC}_{p,q}\ \bigvee_{i \in [0,r)} \Phi_i\right)(x, y)$. Therefore there are $m_0, \ldots, m_s \in |w'|$ such that all of

$$(w', \beta, m_0/p, m_1/q), \ldots, (w', \beta, m_{s-1}/p, m_s/q)$$

satisfy $\bigvee_{i\in[0,r)} \Phi_i$. We have Delilah mark exactly these positions. In response to this, Samson picks $j \in [0, s)$ and places p on position m_j of w' and q on position m_{j+1} of w'. Since $(w', \beta, m_j/p, m_{j+1}/q) \models \bigvee_{i\in[0,r)} \Phi_i$, there is an $i \in [0, r)$ such that $(w', \beta, m_j/p, m_{j+1}/q) \models \Phi_i$, and hence $(w, a, \ell_i/p, \ell_{i+1}/q) \models \Phi_i$. We have Delilah place her pebbles p and q on positions ℓ_i and ℓ_{i+1} of w, respectively. Since Φ_i is a complete description of all formulas satisfied by $(w, a, \ell_i/p, \ell_{i+1}/q)$, the inductive hypothesis applies and Delilah has a winning strategy for the remaining game. □

Definition 4.3.21. Let $n \in \mathbb{N}$, $k \in \mathbb{N}$, and let w and w' be finite words. The game $\mathrm{rFO}_n[\mathrm{FTC}_k](w, w')$ is a variation of the $\mathrm{FO}_n[\mathrm{FTC}_k](w, w')$ game with the following restrictions.

- Initially, one pebble pair is placed on the first positions of the structures, and another pebble pair is placed on the last positions of the structures.
- There are exactly three pairs of pebbles.
- In the existential move on x, that pebble is only allowed to be placed at a position in between the pebbles y and z.
- In the FTC move, after Samson chooses x and y, he removes the other pebble pair z from the board.
- After the FTC move, only the two pebble pairs placed in that move remain on the board.

Theorem 4.3.22. Let w and w' be finite words that end with the same letter, and let $k, n \in \mathbb{N}$. Delilah has a winning strategy in the $\mathrm{rFO}_n[\mathrm{FTC}_k](w, w')$ game iff $(w, 1, \|w\|)$ and $(w', 1, \|w'\|)$ agree on all $\mathrm{rFO}_n[\mathrm{FTC}_k]$ formulas with free variables x and y.

Proof. As in the proof of Theorem 4.3.20, we use induction to prove a stronger claim that implies this theorem.

We claim that for all 2-configurations (α, β), Delilah has a winning strategy for the $\mathrm{rFO}_n[\mathrm{FTC}_k](w, w')$ game in configuration (α, β) iff (w, α) and (w', β) agree on all $\mathrm{rFO}_n[\mathrm{FTC}_k]$ formulas in exactly the free variables from the domains of α and β.

Suppose that there is a formula $\varphi(x, y) \in \mathrm{rFO}_n[\mathrm{FTC}_k]$ with x and y from the domains of α and β such that $(w, \alpha) \models \varphi(x, y)$ and $(w', \beta) \not\models \varphi(x, y)$. If $\varphi(x, y)$ is atomic, $\neg\psi(x, y)$, or $\psi(x, y) \vee \xi(x, y)$, then we argue as in the proof of Theorem 4.3.20. If $\varphi = \exists z\,(x \leq z \leq y \wedge \psi(x, z) \wedge \xi(z, y))$, we have Samson place his pebble z on a position $i \in [\alpha(x), \alpha(y)]$ on w such that $(w, \alpha, i/z) \models \psi(x, z) \wedge \xi(z, y)$. Regardless of Delilah's reply $j \in [\beta(x), \beta(y)]$, we have $(w', \beta, j/z) \not\models \psi(x, z) \wedge \xi(z, y)$. Thus we have either $(w', \beta, j/z) \not\models \psi(x, z)$ or $(w', \beta, j/z) \not\models \xi(z, y)$. We apply the inductive hypothesis to see that Delilah does not have a winning strategy for the remaining game. Finally, if $\varphi = (\mathrm{FTC}_{u,v}\ \psi(u, v))(x, y)$, the same argument as in the proof of 4.3.20 applies because $\psi(u, v))$ has only two free variables.

For the other direction, suppose that (w, α) and (w', β) agree on all $\mathrm{rFO}_n[\mathrm{FTC}_k]$ formulas whose free variables are from the domains of α and β. If Samson's first move in the game on (w, w') in configuration (α, β) is an existential move, placing pebble x on position $i \in [\alpha(x), \alpha(y)]$ on $|w|$, then we let Φ be the conjunction of all finitely many inequivalent $\mathrm{rFO}_{n-1}[\mathrm{FTC}_k]$ formulas satisfied by $(w, \alpha, i/x)$. Thus $(w, \alpha) \models \exists x\, \Phi$, and by our assumption $(w', \beta) \models \exists x\, \Phi$. We have Delilah place her pebble x at $j \in |w'|$ such that $(w', \beta, j/x) \models \Phi$. Since Φ is a complete description of all the formulas satisfied by $(w, \alpha, i/x)$, the inductive hypothesis applies and Delilah has a winning strategy for the remainder of the game.

If Samson's first move is an FTC move. We argue as in the proof of Theorem 4.3.20. □

4.3.2 Transitive Closure with Booleans

Instead of using nested transitive closure operators to simulate the Kleene star operators from the regular expression, it is possible to use just one single transitive closure operator if we allow boolean variables.

Definition 4.3.23. $\mathrm{FO}[\mathrm{TC}, \mathrm{bool}]$ is the extension of $\mathrm{FO}[\mathrm{TC}]$ with boolean variables in the transitive closure operators. These transitive closure formulas are of the form $(\mathrm{TC}_{u,\overline{d},v,\overline{e}}\ \varphi(\overline{d},\overline{e}))(x,\overline{b},y,\overline{c})$ where $\overline{d}$ and $\overline{e}$ are vectors of boolean variables, and $\overline{b}$ and $\overline{c}$ are vectors of boolean values, all of the same finite length.

For semantics, let $w \in \Sigma^+$. Then

$$w \models (\mathrm{TC}_{u,\overline{d},v,\overline{e}}\ \varphi(\overline{d},\overline{e}))(x,\overline{b},y,\overline{c})$$

iff there are $n \in \mathbb{N}$ and $\ell_0,\dots,\ell_n \in |w|$ and vectors of boolean values $\overline{b_0},\dots,\overline{b_n}$ where $\ell_0 = x^w$, $\overline{b_0} = \overline{b}$, $\ell_n = y^w$ and $\overline{b_n} = \overline{c}$ such that for all $i \in [0,n)$, $(w, \ell_i/u, \ell_{i+1}/v) \models \varphi(\overline{b_i}, \overline{b_{i+1}})$.

Theorem 4.3.24. $\mathrm{FO}[\mathrm{TC}_1, \mathrm{bool}]$ on finite words captures exactly the regular languages. Moreover, if n is the number of states of an automaton (possibly nondeterministic) that accepts the language, then at most $\lceil \log n \rceil$ booleans are necessary.

Proof. Let L be a regular language, and $A = (Q,\Sigma,q_0,\delta,F)$ a finite automaton such that $\mathcal{L}(A) = L$. We write $\langle q \rangle$ for a fixed binary encoding of state q into $\lceil \log |Q| \rceil$ boolean variables. We construct a formula $\psi_A(x,\overline{b},y,\overline{c})$ that walks through the automaton for just one step, where the automaton transitions from state $\langle \overline{b} \rangle$ to state $\langle \overline{c} \rangle$ when reading the symbol at position x. The variable y is used to force the transitive closure operator to process one symbol at a time, except for the last position, where there is no next position to move to.

$$\psi_A(x, \overline{b}, y, \overline{c}) = (\mathrm{Suc}(x, y) \vee x = y = \max) \wedge \bigvee_{\delta(q,a)=q'} \left(\overline{b} = \langle q \rangle \wedge Q_a(x) \wedge \overline{c} = \langle q' \rangle\right)$$

$$\varphi_A = \bigvee_{q_f \in F} \left(\mathrm{TC}_{x,\overline{b},y,\overline{c}}\ \psi_A(x, \overline{b}, y, \overline{c})\right) (\min, \langle q_0 \rangle, \max, \langle q_f \rangle)$$

It is easy to see that $w \in \mathcal{L}(A)$ iff $w \models \varphi_A$. For the other direction, we show how to express the TC operator with a sequence of universal monadic second-order quantifiers, one for each possible setting of the boolean variables.

$$w \models \left(\mathrm{TC}_{x,\overline{b},y,\overline{c}}\ \varphi\right) (x, \overline{b}, y, \overline{c})$$

$$\text{iff}$$

$$w \models \forall X_{\overline{0}} \ldots \forall X_{\overline{1}} \left(X_{\overline{b}}(x) \wedge \forall x \forall y \bigwedge_{\overline{b},\overline{c}} ((X_{\overline{b}}(x) \wedge \varphi(x, \overline{b}, y, \overline{c})) \to X_{\overline{c}}(y)) \right) \to X_{\overline{c}}(y)$$

Using Theorem 4.3.8, it follows that the language described by the transitive closure formula is regular. □

Open Problem 4.3.25. What is the minimum number of boolean variables required for a FO[TC_1, bool] formula to express a given regular language? Can we do better than the number-of-states upper bound from the previous theorem?

Any lower bound on the number of booleans would imply that the language is not in FO[TC_1], thus showing that there is a language of generalized star-height at least 2.

Open Problem 4.3.26. Investigate the trade-off between the number of booleans and TC depth.

4.4 Three Candidate Languages and Short Tile Machines

We present a sequence of candidates for languages of generalized star-height 2. While still relatively simple, our first candidate language requires a form of staged counting that is different from the counting used in the languages $W(h,k,r,m)$ that Thomas proved to have generalized star-height at most 1 [43]. Let $\Sigma_2 = \{\mathtt{a},\mathtt{b}\}$, and consider the language L_1 of all strings with an even number of $\mathtt{b}$'s that occur after an even number of $\mathtt{a}$'s.

$$L_1 := \mathcal{L}\left((\mathtt{ab}^\star\mathtt{a} \cup \mathtt{b}(\mathtt{ab}^\star\mathtt{a})^\star\mathtt{b})^\star\right)$$

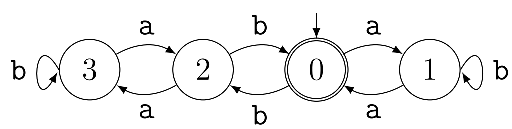

Figure 4.2. Finite automaton for the language L_1.

Unfortunately, the syntactic monoid of L_1 is nilpotent of class 2, and thus the techniques from [30] apply. For illustrative purposes, we explicitly construct a regular expression R of generalized star-height 1 for L_1, not unlike the construction from Thomas [43].

$$T := \mathtt{b} \cup \mathtt{ab}^\star\mathtt{a}$$

$$E := (\mathtt{b} \cup (\mathtt{ab}^\star\mathtt{a})\mathtt{b}^\star(\mathtt{ab}^\star\mathtt{a}))^\star$$

$$O := \mathtt{b}^\star(\mathtt{ab}^\star\mathtt{a})E$$

$$R := ((TT)^\star \cap E) \cup (T(TT)^\star \cap O)$$

The expression T matches the two critical patterns $\mathtt{b}$ and $\mathtt{ab}^\star\mathtt{a}$ in the string. We can count the number of occurrences of these patterns with the expressions $(TT)^\star$ for an even number and $T(TT)^\star$ for an odd number. The expression E checks whether the pattern $\mathtt{ab}^\star\mathtt{a}$ occurs an even number of times (or equivalently, whether the number of a's is a multiple of 4). Similarly, O checks whether this pattern occurs an odd number of times, which is equivalent to checking that the number of a's is congruent 2 modulo 4.

Our second candidate language L_2 has been proposed in [30], and it is very similar to our first candidate L_1 – we only replace the pattern b with $(ba^\star b)$.

$$L_2 := \mathcal{L}((ab^\star a \cup ba^\star b(ab^\star a)^\star ba^\star b)^\star)$$

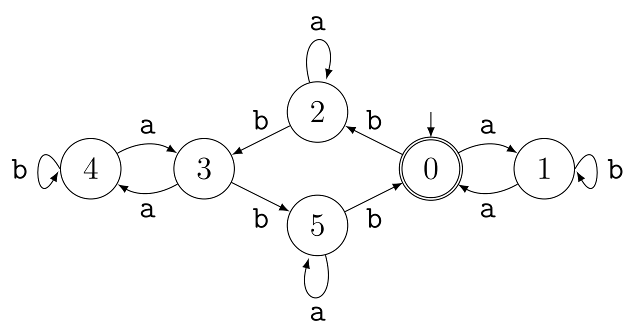

Figure 4.3. Finite automaton for the language L_2.

The syntactic monoid of this language is a group with 48 elements that is not nilpotent, and we do not know whether it has generalized star-height 2. In our attempts to prove that L_2 has generalized star-height 2, we initially focused on only those strings from Σ_2 where all b's are "far apart", meaning that in between any two b's there are at least four a's. Since aa in $M(L_2)$ is the identity, it appeared to us that the counting problem that is inherent to L_2 should not be easier to solve on this subset than on all of $\Sigma_2^\star$. However, we discovered that L_2 on this restricted subset has generalized star-height 1. In the following construction, we use regular expressions T to count the total number of the two patterns ($\mathtt{ab}^\star\mathtt{a}$) and ($\mathtt{ba}^\star\mathtt{b}$), and expressions O and E that determine the parity of the number of ($\mathtt{ab}^\star\mathtt{a}$) patterns. A regular expression R for L_2 is then easily constructed as a boolean combination of the former expressions.

$$T := \mathtt{ab^\star a} \cup \mathtt{ba^\star b}$$

$$E := (\mathtt{aaaa} \cup \mathtt{aaba^\star baa} \cup \mathtt{ba^\star b} \cup \mathtt{abaaa} \cup \mathtt{aaaba})^\star$$

$$O := E(\mathtt{aba} \cup \mathtt{aa}(\mathtt{ba^\star b})^\star)$$

$$R := ((TT)^\star \cap E) \cup ((TT)^\star T \cap O)$$

As an alternative candidate language we developed L_3, which features a different kind of staged counting process. As in L_1, we count the number of certain $\mathtt{b}$'s, but for L_3 whether a $\mathtt{b}$ counts not only depends on the number of $\mathtt{a}$'s seen since the last relevant $\mathtt{b}$, but the counting modulus also flips back and forth between 2 and 3.

$$L_3 = \mathcal{L}((\mathtt{ab^\star a} \cup \mathtt{b}(\mathtt{ab^\star ab^\star a})^\star \mathtt{b})^\star)$$

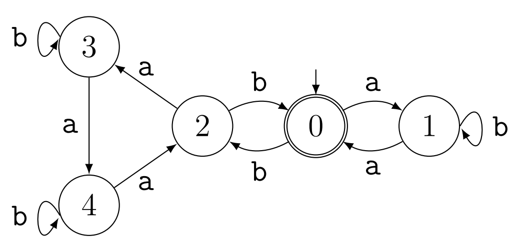

Figure 4.4. Finite automaton for the language L_3.

The syntactic monoid of L_3 is S_5, the symmetric group of degree five, and thus neither nilpotent nor solvable. Similar to what we observed for L_2, $\mathtt{a}^6$ in $M(L_3)$ is equal to the identity, but again restricting the alphabet to for example all strings with at least six $\mathtt{a}$'s in between $\mathtt{b}$'s allows us to decide L_3 in generalized star-height 1. While initially we hoped to prove that the restricted version of L_3 still has generalized star-height 2, proving that this is not the case has been a cumbersome process, and we feel that we gained some valuable insights along the way.

In particular, we developed the notion of *short tile machines.* These machines are inspired by the kind of pattern construction that was used to decide L_2 on the restricted set of strings. A short tile machine is an automaton where the states are arranged in a circle formed by `a`-transitions, and `b`-transitions are allowed among arbitrary pairs of states. The machines A_1, A_2 and A_3 in Figure 4.5 are examples of short tile machines for inputs where in between any `b`'s there are at least five `a`'s.

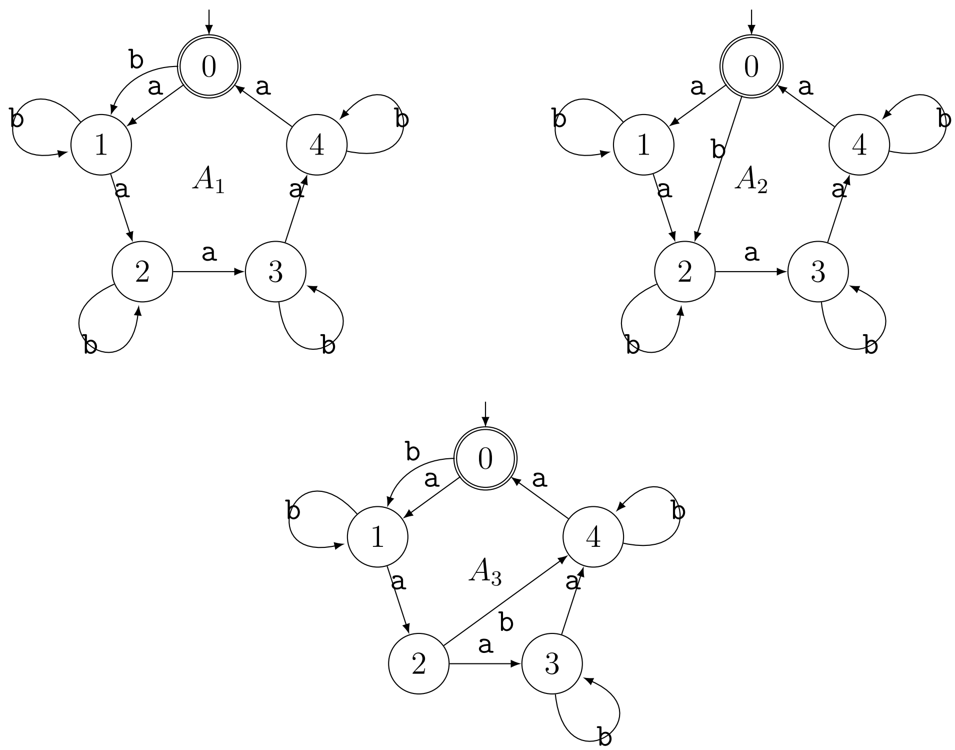

Figure 4.5. Short tile machines A_1, A_2 and A_3.

While all three of these short tile machines are deterministic, there is no reason to require this in general. In order to better understand the power of these machines, we proved that A_3 cannot be accepted by any boolean combination of A_1 and A_2. We tried to extend these ideas and prove that our restricted candidate language L_3

cannot be accepted by any boolean combination of 6-cycle short tile machines, but after several unsuccessful attempts we decided to try to prove the opposite. We wrote a simulation environment for boolean combinations of short tile machines, and discovered that there is a boolean combination of four 6-cycle short tile machines that accepts L_3 on restricted inputs. Since enumerating all possible boolean combinations of 6-cycle short tile machines is a daunting if not infeasible task, we used the following lemma to significantly reduce the number of machines that we need to look at.

We conjecture that $M(L_3)$ is not a homomorphic image of any boolean combination of automata with syntactic monoid $M(L_1)$ and $M(L_2)$, and thus L_3 cannot be recognized by a boolean combination of automata that recognize the first two languages.

Before we investigate this conjecture, we observe that $M(L_1)$ and $M(L_2)$ are isomorphic, and thus any boolean combination of these automata has a syntactic monoid that is isomorphic to $M(L_1)$. We define the homomorphism $\varphi : \Sigma^\star \to \Sigma^\star$ with $\varphi(\mathtt{a}) := \mathtt{aa}$ and $\varphi(\mathtt{b}) = \mathtt{b}$, and we observe that $L_2 = \varphi^{-1}(L_1)$, or equivalently for every $w \in \Sigma^\star$, $w \in L_1$ iff $\varphi(w) \in L_2$. Even more so, the following lemma holds.

Claim 4.4.1. For every $q \in Q_1$, let $\langle q \rangle \in Q_2$ such that $\langle q \rangle \equiv 2q \pmod 5$. For every $w \in \Sigma^\star$ and every $q \in Q_1$, $\delta_2^\star(\langle q \rangle, \varphi(w)) \equiv \langle \delta_1^\star(q, w) \rangle$.

Proof. We use induction on the string w. For $w = \varepsilon$, we have

$$\delta_2^\star(\langle q \rangle, \varphi(\varepsilon)) = \delta_2^\star(\langle q \rangle, \varepsilon) = \langle q \rangle = \langle \delta_1^\star(q, \varepsilon) \rangle \quad .$$

For $w = ux$ where $u \in \Sigma^\star$ and $x \in \Sigma$, we have $\varphi(x) \in \{\mathtt{aa}, \mathtt{b}\}$ and thus

$$\begin{aligned} \delta_2^\star(\langle q \rangle, \varphi(ux)) = \delta_2^\star(\langle q \rangle, \varphi(u)\varphi(x))) &= \delta_2^\star(\delta_2^\star(\langle q \rangle, \varphi(u)), \varphi(x)) \\ &= \delta_2^\star(\langle \delta_1^\star(q, u) \rangle, \varphi(x)) \\ &= \langle \delta_1^\star(\delta_1^\star(q, u), x) \rangle = \langle \delta_1^\star(q, ux) \rangle \quad . \qquad \square \end{aligned}$$

We define the map $\psi : M(L_1) \to M(L_2)$ with $\psi([w]_{L_1}) = [\varphi(w)]_{L_2}$ for $w \in \Sigma^\star$, and claim that it is well-defined and an isomorphism. Before we prove that ψ is well-defined, we argue that every word from $\Sigma^\star$ is L_2-equivalent to a word from $\varphi(\Sigma^\star)$. To this end we define the map $\tau : \Sigma^\star \to \varphi(\Sigma^\star)$ with $\tau(\varepsilon) = \varepsilon$, and for $w \in \Sigma^\star$ and $i \in \mathbb{N}$,

$$\tau(w\mathtt{ba}^i) := \begin{cases} \tau(w)\mathtt{ba}^i & \text{if } i \text{ is even} \\ \tau(w)\mathtt{ba}^{i+5} & \text{if } i \text{ is odd} \end{cases} .$$

With this definition, every block of $\mathtt{a}$'s in $\tau(w)$ has even length and thus is in the image of φ. To see that $w \sim_{L_2} \tau(w)$, we only need to observe that $\mathtt{a}^5$ acts as the identity in $M(L_2)$.

$$\begin{tikzcd} \Sigma^\star \arrow[r, "\varphi"] \arrow[d, "\varphi_{L_1}"] & \Sigma^\star \arrow[d, "\varphi_{L_2}"] \\ M(L_1) \arrow[r, "\psi"'] & M(L_2) \end{tikzcd}$$

Figure 4.6. Maps for the construction of an isomorphism from $M(L_1)$ to $M(L_2)$.

For the well-definedness of ψ, let $x, y \in \Sigma^\star$ with $x \sim_{L_1} y$. Using the above lemma, we have that for all $w, w' \in \Sigma^\star$,

$$\begin{aligned} w\varphi(x)w' \in L_2 \Leftrightarrow \tau(w)\varphi(x)\tau(w') \in L_2 &\Leftrightarrow \varphi^{-1}(\tau(w)\varphi(x)\tau(w')) \in L_1 \\ &\Leftrightarrow \varphi^{-1}(\tau(w))\ x\ \varphi^{-1}(\tau(w')) \in L_1 \\ &\Leftrightarrow \varphi^{-1}(\tau(w))\ y\ \varphi^{-1}(\tau(w')) \in L_1 \\ &\Leftrightarrow \varphi^{-1}(\tau(w)\varphi(y)\tau(w')) \in L_1 \\ &\Leftrightarrow \tau(w)\varphi(y)\tau(w') \in L_2 \Leftrightarrow w\varphi(y)w' \in L_2 \quad . \end{aligned}$$

Thus $\varphi(x) \sim_{L_2} \varphi(y)$, $[(\varphi(x)]_{L_2} = [\varphi(y)]_{L_2}$.

To see that ψ is a homomorphism, we observe that for arbitrary $x, y \in \Sigma^\star$,

$$\begin{aligned}\psi([x]_{L_1})\psi([y]_{L_1}) = [\varphi(x)]_{L_2}[\varphi(y)]_{L_2} = [\varphi(x)\varphi(y)]_{L_2} = [\varphi(xy)]_{L_2} &= \psi([xy]_{L_1}) \\ &= \psi([x]_{L_1}[y]_{L_2}) \quad .\end{aligned}$$

To see that ψ is a bijection, suppose that $\psi([x]_{L_1}) = \psi([y]_{L_1})$ for some $x, y \in \Sigma^\star$. Thus $[\varphi(x)]_{L_2} = [\varphi(y)]_{L_2}$, and for every $w, w' \in \Sigma^\star$ we have

$$\begin{aligned}wxw' \in L_1 \Leftrightarrow \varphi(wxw') \in L_2 &\Leftrightarrow \varphi(w)\varphi(x)\varphi(w') \in L_2 \\ &\Leftrightarrow \varphi(w)\varphi(y)\varphi(w') \in L_2 \\ &\Leftrightarrow \varphi(wyw') \in L_2 \Leftrightarrow wyw' \in L_1 \quad .\end{aligned}$$

Definition 4.4.2. A *finite semiautomaton* is a tuple (Σ, Q, δ), where Σ is a finite alphabet, Q is a finite set of states, and $\delta : (Q \times \Sigma) \to Q$ is a transition function. An *instantiation* of a semiautomaton $A = (\Sigma, Q, \delta)$ is a finite automaton $A[q_0, F] = (\Sigma, Q, \delta, q_0, F)$ where $q_0 \in Q$ and $F \subseteq Q$.

Definition 4.4.3. Let A, B and C be monoids, and let $\varphi : A \to B$ and $\psi : A \to C$ be homomorphisms. We say that ψ *factors through* φ if for all $x, y \in A$, $\varphi(x) = \varphi(y)$ implies $\psi(x) = \psi(y)$.

Lemma 4.4.4. Let A be a semiautomaton over Σ, and let $L \subseteq \Sigma^\star$. The language L is recognized by a boolean combination of instantiations of A iff the syntactic morphism φ_L of L factors through the transition morphism φ_A of A.

Proof. For the forward direction, we first observe that the transition monoid of any boolean combination of instantiations of A is still $M(A)$. Invoking basic results from algebraic language theory, we know that $M(L)$ divides $M(A)$, i.e. there is a subset $M \subseteq M(A)$ and a homomorphism $\psi : M \to M(L)$ such that $\varphi_L = \psi \circ \varphi_A$. Thus φ_L factors through φ_A.

For the backward direction, let $M(L)$ be the syntactic monoid of L. Using basic results from algebraic language theory, we have a set $F \subseteq M(L)$ such that $L = \varphi_L^{-1}(F)$. Using the relationship between φ_A and φ_L, we observe that we can also express L in terms of the transition monoid of A, namely $L = \varphi_A^{-1}(F')$ where $F' = \varphi_A(L)$. To see that this is true, let $w \in L$. Then $\varphi_A(w) \in \varphi_A(L) = F'$ and thus $w \in \varphi_A^{-1}(F')$. Conversely, suppose $w \in \varphi_A^{-1}(F')$. Then $\varphi_A(w) \in F' \in \varphi_A(L)$. Thus there is a $u \in L$ such that $\varphi_A(w) = \varphi_A(u)$, hence $\varphi_L(w) = \varphi_L(u)$ and $w \in L$. Now that we have $L = \varphi_A^{-1}(L)$, we construct a boolean combination of instantiations of A to recognize L as follows.

$$A_L := \bigvee_{x \in F'} \bigwedge_{q \in Q(A)} A[q, qx] \qquad \square$$

4.5 Modular Counting of Substrings

Definition 4.5.1. Let $p \in \Sigma^+$ and $w \in \Sigma^\star$. The number of occurrences of p as a substring of w is $\#_p(w) := |\{1 \leq i \leq \|w\| - \|p\| + 1 \mid w_i \dots w_{i+\|p\|-1} = p\}|$.

Definition 4.5.2. Let $p \in \Sigma^+$. A strict suffix s of p is a *repeating suffix* of p if p is a suffix of ps.

For example, the string `abaabaab` has exactly two repeating suffixes: `aab` and `aabaab`. The string `babbab` also has exactly two repeating suffixes: `bab` and `abbab`. The string `bbbbbabbbbb` has exactly five repeating suffixes.

Proposition 4.5.3. Let $p \in \Sigma^+$, let $k, m \in \mathbb{N}$, $k < m$, and define the language

$$L_p^{k,m} := \{w \in \Sigma^\star \mid \#_p(w) \equiv k \pmod{m}\} \quad .$$

The generalized star-height of $L_p^{k,m}$ is 1.

Proof. If the pattern p does not have any repeating suffixes, then we construct a regular expression E of generalized star-height 1 as follows.

$$E := T\ ((pT)^m)^\star\ (pT)^k$$
$$T := \overline{\Sigma^\star p \Sigma^\star}$$

Every occurrence of p can only be matched by the subexpression p in E, and every such match corresponds to a unique occurrence of p, thus $\mathcal{L}(E) = L_p^{k,m}$.

We extend this construction to patterns with repeating suffixes. Let S_p be the set of all repeating suffixes of p that do not have another repeating suffix as their prefix. For example, $S_{\texttt{abaabaab}}$ only contains `aab` and not `aabaab`, since `aab` is a prefix of `aabaab`.

In the following expressions, we write S for the union over all strings from S_p.

$$E := \begin{cases} U\ ((S \cup V)^m)^\star\ (S \cup V)^{k-1}\ W & \text{if } k > 0 \\ U\ ((S \cup V)^m)^\star\ (S \cup V)^{m-1}\ W \cup \overline{\Sigma^\star p \Sigma^\star} & \text{if } k = 0 \end{cases}$$
$$U := \overline{\Sigma^\star p \Sigma^+} \cap \Sigma^\star p$$
$$V := \overline{S\Sigma^\star} \cap \overline{\Sigma^\star p \Sigma^+} \cap \Sigma^\star p$$
$$W := \overline{S\Sigma^\star} \cap \overline{\Sigma^\star p \Sigma^\star}$$

The expressions S, U, V and W all have generalized star-height 0, and the expression E has generalized star-height 1. Expression U matches all strings that end with p but do not contain p anywhere else, expression V matches all strings that end with p, do not contain p anywhere else, and also do not start with any repeating suffix. Expression W matches all strings that do not contain p and do not start with any repeating suffix.

We claim that $\mathcal{L}(E) = L_p^{k,m}$. To see that $\mathcal{L}(E) \subseteq L_p^{k,m}$ for $k > 0$, we observe that the initial U matches exactly one occurrence of p. Every substring matched by

$S \cup V$ increases the number of occurrences of p by exactly one, either generating a new occurrence with a repeating suffix, or not starting with a generating suffix and ending with p. The part of the string matched so far, including the expression $S \cup V$, also always ends with p. Finally, W does not contribute any additional occurrences of p. To argue $L_p^{k,m} \subseteq \mathcal{L}(E)$, we note that the first occurrence of p is matched by U, and any following occurrence is matched by $S \cup V$. For $k = 0$, this argument remains unchanged, except that we also need to consider the case where there are no occurrences of p at all. The strings without any occurrences of p are matched by $\overline{\Sigma^\star p \Sigma^\star}$. □

4.6 Word Problems for Symmetric Groups

Consider the language L accepted by the automaton in Figure 4.7, with state 0 as the initial and final state. All expressions in this section can be easily modified to work for any choice of initial and final states.

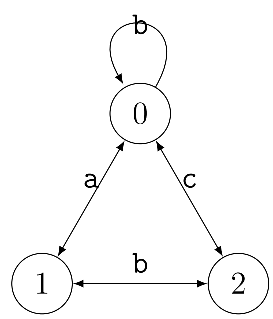

Figure 4.7. Finite semiautomaton for the language L.

This language is motivated by a version of the word problem of S_3. The variant with loops on states 1 and 2 appears to be more complicated, and we currently do not have an expression of generalized star-height 1 to match that language even with just two loops.

We can easily write an expression of generalized star-height 2 for L as follows.

$$(\mathtt{b} \cup \mathtt{a}(\mathtt{bb})^\star\mathtt{a} \cup \mathtt{ab}(\mathtt{bb})^\star\mathtt{c} \cup \mathtt{c}(\mathtt{bb})^\star\mathtt{c} \cup \mathtt{cb}(\mathtt{bb})^\star\mathtt{a})^\star$$

Before we present an expression of generalized star-height 1, we make two observations about this language.

Claim 4.6.1. In any string that transitions from state 0 to state 0, the number of $\mathtt{a}$'s plus the number of $\mathtt{c}$'s is even. In any string that transitions from state 0 to state 1 or 2, this number is odd.

Proof. To transition from state 0 to state 0 without using state 0 as an intermediate state, a string has to be equal to $\mathtt{b}$, or start and end with $\mathtt{a}$ or $\mathtt{c}$ but not contain any other $\mathtt{a}$'s or $\mathtt{c}$'s. □

We can easily write a regular expression E for all strings where the total number of $\mathtt{a}$'s and $\mathtt{c}$'s is even.

$$E := \mathtt{b}^\star((\mathtt{a} \cup \mathtt{c})\mathtt{b}^\star(\mathtt{a} \cup \mathtt{c})\mathtt{b}^\star)^\star$$

Claim 4.6.2. A string w transitions from state 0 to the dead state without using the dead state as an intermediate state iff it is of the form uv, where $u \in \mathcal{L}(E)$, $v \in \mathcal{L}(X)$, and

$$X := \mathtt{a}(\mathtt{bb})^\star\mathtt{c} \cup \mathtt{ab}(\mathtt{bb})^\star\mathtt{a} \cup \mathtt{c}(\mathtt{bb})^\star\mathtt{a} \cup \mathtt{cb}(\mathtt{bb})^\star\mathtt{c} \quad .$$

Proof. Let w be a string that transitions from state 0 to the dead state without any intermediate use of the dead state. Let u be a (possibly empty) prefix of w up to the last occurrence of state 0 in the corresponding state transition sequence, and let v be the remainder of w. By the previous claim, u is in $\mathcal{L}(E)$. To transition away from state 0, v has to start with $\mathtt{a}$ or $\mathtt{c}$. For each of these two cases, there are exactly two ways to transition to the dead state without using state 0, as listed in the expression X.

It is straightforward to verify that every string from $\mathcal{L}(EX)$ transitions from state 0 to the dead state. □

Combining these two claims, we can now write a regular expression R of generalized star-height 1 for all of L.

$$R := E \cap \overline{EX\Sigma^\star}$$

Proposition 4.6.3. $\mathcal{L}(R) = L$.

Proof. Let $w \in L$, i.e. w transitions from state 0 to state 0. By Claim 4.6.1 we have $w \in \mathcal{L}(E)$, and by claim 4.6.2, w cannot have any string from $\mathcal{L}(EX)$ as a prefix, thus $w \notin \mathcal{L}(EX\Sigma^\star)$.

Conversely, suppose that $w \in \mathcal{L}(R)$. Since $w \notin \mathcal{L}(EX\Sigma^\star)$, w does not have a prefix from $\mathcal{L}(EX)$, and thus by Claim 4.6.2 we know that starting at state 0, w does not transition to the dead state. Since $w \in \mathcal{L}(E)$, the second part of Claim 4.6.1 implies that w cannot transition to states 1 or 2. Thus w has to transition back to state 0, and $w \in L$. $\square$

Very similar ideas can be used to construct expressions of generalized star-height 1 for more restricted versions of the language L, but we currently do not know how to write such an expression for the variant of L with a second loop.

CONCLUSION

In the first half of this dissertation, we presented a series of new results on first-order logic with two variables on finite words. We provided a new and complete characterization of the properties expressible in FO^2 on finite words, proved that the quantifier alternation hierarchy for this logic is strict, and settled the main remaining question about the complexity of the satisfiability problem for this logic. We feel that these results complete our understanding of the expressiveness of FO^2 on finite words. Nevertheless, interesting questions remain for FO^2 on more general structures such as infinite words [10] and trees [2,3]. We restricted our attention to finite words, but suspect that many of our results from the first two chapters generalize to infinite words.

The second half of this dissertation is more exploratory in nature. We gained new insights into both the succinctness of first-order logic and the generalized star-height hierarchy, and developed promising techniques that we hope will advance our understanding of both problems. Many questions remain open. Our grasp of succinctness is still very basic, and while we believe that the techniques developed here will lead to new and improved lower bounds on succinctness for certain special cases, we feel that further progress depends on the development of fundamentally new approaches and methods that currently are beyond our reach. Many of these questions on succinctness are closely related to some of the main open problems in computational complexity, most importantly the trade-off between parallel time and hardware. For the generalized star-height hierarchy, we are more optimistic and believe that our new logical approach might lead to further progress in the near future.

BIBLIOGRAPHY

[1] Adler, Micah, and Immerman, Neil. An $n!$ lower bound on formula size. *ACM Transactions on Computational Logic 4*, 3 (2003), 296–314.

[2] Bojańczyk, Mikołaj. Two-way unary temporal logic over trees. In *IEEE Symposium on Logic in Computer Science* (2007), pp. 121–130.

[3] Bojańczyk, Mikołaj, and Segufin, Luc. Tree languages defined in first-order logic with one quantifier alternation. *Logical Methods in Computer Science 6*, 4:1 (2010).

[4] Brzozowski, Janusz A., and Knast, Robert. The dot-depth hierarchy of star-free languages is infinite. *Journal of Computer and System Science 16* (1978), 37–55.

[5] Büchi, J. Richard. Weak second-order arithmetic and finite automata. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik 6* (1960), 66–92.

[6] Büchi, J. Richard, and Elgot, Calvin C. Decision problems of weak second order arithmetics and automata, part I. *Notices of the American Mathematical Society 5* (1958), 834.

[7] Calo, Ariel, and Makowsky, Johann A. The Ehrenfeucht-Fraïssé games for transitive closure. In *International Symposium on Logical Foundations of Computer Science* (1992), pp. 57–68.

[8] Eggan, Lawrence C. Transition graphs and the star-height of regular events. *Michigan Mathematical Journal 10*, 4 (1963), 385–397.

[9] Etessami, Kousha, Vardi, Moshe Y., and Wilke, Thomas. First-order logic with two variables and unary temporal logic. In *IEEE Symposium on Logic in Computer Science* (1997), pp. 228–235.

[10] Etessami, Kousha, Vardi, Moshe Y., and Wilke, Thomas. First-order logic with two variables and unary temporal logic. *Information and Computation 179*, 2 (2002), 279–295.

[11] Fürer, Martin. The computational complexity of the unconstrained limited domino problem (with implications for logical decision problems). In *Logic and Machines* (1984), Egon Börger, Gisbert Hasenjaeger, and Dieter Rödding, Eds., vol. 171 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, pp. 312–319.

[12] Glasser, Christian, Schmitz, Heinz, and Selivanov, Victor L. Efficient algorithms for membership in boolean hierarchies of regular languages. In *Symposium on Theoretical Spects of Computer Science* (2008), pp. 337–348.

[13] Grädel, Erich. On transitive closure logic. In *Computer Science Logic* (1991), pp. 149–163.

[14] Grohe, Martin, and Schweikardt, Nicole. Comparing the succinctness of monadic query languages over finite trees. *RAIRO Theoretical Informatics and Applications 51*, 1 (2004), 74–113.

[15] Grohe, Martin, and Schweikardt, Nicole. The succinctness of first-order logic on linear orders. *Logical Methods in Computer Science 1*, 1:6 (2005).

[16] Gruber, Hermann, and Holzer, Markus. Finite automata, digraph connectivity, and regular expression size. In *International Colloquium on Automata, Language and Programming* (2008), pp. 39–50.

[17] Hashiguchi, Kosaburo. Representation theorems on regular languages. *Journal of Computer and System Sciences 27*, 1 (1983), 101–115.

[18] Hashiguchi, Kosaburo. Algorithms for determining relative star height and star height. *Information and Computation 78*, 2 (1988), 124–169.

[19] Immerman, Neil. Languages that capture complexity classes. *SIAM Journal of Computing 16*, 4 (1987), 760–778.

[20] Immerman, Neil. DSPACE[n^k] = VAR[$k+1$]. In *IEEE Structure in Complexity Theory Symposium* (1991).

[21] Immerman, Neil. *Descriptive Complexity.* Springer-Verlag, New York, NY, 1999.

[22] Immerman, Neil, Buss, Jonathan F., and Barrington, David A. Mix. Number of variables is equivalent to space. *Journal of Symbolic Logic 66*, 3 (2001), 1217–1230.

[23] Immerman, Neil, and Kozen, Dexter. Definability with bounded number of bound variables. *Information and Computation 83*, 2 (1989), 121–139.

[24] Kamp, Johan A.W. *Tense logic and the theory of linear order.* PhD thesis, University of California, Los Angeles, 1968.

[25] Karchmer, Mauricio, and Wigderson, Avi. Monotone circuits for connectivity require super-logarithmic depth. *SIAM Journal of Discrete Mathematics 3*, 2 (1990), 255–265.

[26] Lewis, Harry R. Complexity results for classes of quantificational formulas. *Journal of Computer and System Sciences 21* (1980), 317–353.

[27] McNaughton, Robert. The loop complexity of regular events. *Information Sciences 1* (1969), 305–328.

[28] McNaughton, Robert, and Papert, Seymour A. *Counter-free Automata*. MIT Press, Cambridge, MA, 1971.

[29] Pin, Jean-Eric. *Varieties of Formal Languages*. North Oxford Academic, London and Plenum, New York, NY, 1986.

[30] Pin, Jean-Eric, Straubing, Howard, and Thérien, Denis. Some results on the generalized star-height problem. *Information and Computation 101*, 2 (1992), 219–250.

[31] Pin, Jean-Eric, and Weil, Pascal. Polynomial closure and unambiguous product. *Theory of Computing Systems 30* (1997), 1–39.

[32] Potthoff, Andreas. *Logische Klassifizierung regulärer Baumsprachen*. PhD thesis, Institut für Informatik und Praktische Mathematik, Universität Kiel, 1994.

[33] Schützenberger, Marcel P. Sur le produit de concatenation non ambigu. *Semigroup Forum 13* (1976), 47–75.

[34] Schwentick, Thomas, Thérien, Denis, and Vollmer, Heribert. Partially-ordered two-way automata: a new characterization of DA. In *Developments in Language Theory* (2001), pp. 239–250.

[35] Stockmeyer, Larry J. *The complexity of decision problems in automata theory*. PhD thesis, MIT, 1974.

[36] Straubing, Howard. *Finite Automata, Formal Logic, and Circuit Complexity*. Birkhäuser, Boston, MA, 1994.

[37] Straubing, Howard, and Thérien, Denis. Weakly iterated block products. In *Latin American Theoretical Informatics Conference* (2002), pp. 91–104.

[38] Straubing, Howard, and Weil, Pascal. On a conjecture concerning dot-depth two languages. *Theoretical Computer Science 104* (1992), 161–183.

[39] Tesson, Pascal, and Thérien, Denis. Diamonds are forever: the variety DA. In *Semigroups, Algorithms, Automata and Languages* (Coimbra, Portugal, 2001), World Scientific, pp. 475–500.

[40] Tesson, Pascal, and Thérien, Denis. Algebra meets logic: the case of regular languages. *Logical Methods in Computer Science 3*, 1:4 (2007).

[41] Thérien, Denis. Subword counting and nilpotent groups. In *Combinatorics on Words: Progress and Perspectives*, L. J. Cummings, Ed. Academic Press, Orlando, FL, 1983, pp. 297–305.

[42] Thérien, Denis, and Wilke, Thomas. Over words, two variables are as powerful as one quantifier alternation. In *ACM Symposium on Theory of Computing* (1998), pp. 234–240.

[43] Thomas, Wolfgang. Remark on the star-height-problem. *Theoretical Computer Science 13* (1981), 231–237.

[44] Thomas, Wolfgang. An application of the Ehrenfeucht-Fraïssé game in formal language theory. *Mémoires de la S.M.F. 16* (1984), 11–21.

[45] Trakhtenbrot, Boris A. Finite automata and the logic of monadic predicates. *Doklady Akademii Nauk SSSR 140* (1961), 326–329.

[46] Trakhtenbrot, Boris A., and Barzdin, Janis M. *Finite Automata*. North-Holland, Amsterdam, 1973.

[47] Weis, Philipp, and Immerman, Neil. Structure theorem and strict alternation hierarchy for FO^2 on words. *Logical Methods in Computer Science 5*, 3:4 (2009).

[48] Wilke, Thomas. CTL^+ is exponentially more succinct than CTL. In *Conference on Foundations of Software Technology and Theoretical Computer Science* (1999), pp. 110–121.