Theoretical Computer Science 288 (2002) 129–152
Theoretical Computer Science
www.elsevier.com/locate/tcs

Guarded fixed point logics and the monadic theory of countable trees

Erich Grädel [1]

Mathematische Grundlagen der Informatik, RWTH Aachen, 52056 Aachen, Germany

Abstract

Different variants of guarded logics (a powerful generalization of modal logics) are surveyed and an elementary proof for the decidability of guarded fixed point logics is presented. In a joint paper with Igor Walukiewicz, we proved that the satisfiability problems for guarded fixed point logics are decidable and complete for deterministic double exponential time (E. Grädel and I. Walukiewicz, Proc. 14th IEEE Symp. on Logic in Computer Science, 1999, pp. 45–54). That proof relies on alternating automata on trees and on a forgetful determinacy theorem for games on graphs with unbounded branching. The exposition given here emphasizes the tree model property of guarded logics: every satisfiable sentence has a model of bounded tree width. Based on the tree model property, we show that the satisfiability problem for guarded fixed point formulae can be reduced to the monadic theory of countable trees (SωS), or to the μ-calculus with backwards modalities.

Keywords: Guarded logics; Fixed point logics; Decidability

1. Introduction

Guarded logics are defined by restricting quantification in first-order logic, second-order logic, fixed point logics or infinitary logics in such a way that, semantically speaking, each subformula can 'speak' only about elements that are 'very close together' or 'guarded'.

Syntactically this means that all first-order quantifiers must be relativised by certain 'guard formulae' that tie together all the free variables in the scope of the quantifier. Quantification is of the form

$$\exists \boldsymbol{y}(\alpha(\boldsymbol{x},\boldsymbol{y}) \wedge \psi(\boldsymbol{x},\boldsymbol{y})) \quad \text{or} \quad \forall \boldsymbol{y}(\alpha(\boldsymbol{x},\boldsymbol{y}) \rightarrow \psi(\boldsymbol{x},\boldsymbol{y}))$$

where quantifiers may range over a tuple $\boldsymbol{y}$ of variables, but are 'guarded' by a formula α that must contain all the free variables of the formula ψ that is quantified over. The

E-mail address: graedel@informatik.rwth-aachen.de (E. Grädel).
[1] URL: www-mgi.informatik.rwth-aachen.de

PII: S0304-3975(01)00151-7

guard formulae are of a simple syntactic form (in the basic version, they are just atoms). Depending on the conditions imposed on guard formulae, one has logics with different levels of 'closeness' or 'guardedness'. Again, there is a syntactic and a semantic view of such guard conditions.

Let us start with the logic GF, the *guarded fragment* of first-order logic, as it was introduced by Andréka et al. [1].

Definition 1.1. GF is defined inductively as follows:
(1) Every relational atomic formula $Rx_{i_1} \cdots x_{i_m}$ or $x_i = x_j$ belongs to GF.
(2) GF is closed under boolean operations.
(3) If $\boldsymbol{x}, \boldsymbol{y}$ are tuples of variables, $\alpha(\boldsymbol{x}, \boldsymbol{y})$ is a positive atomic formula, and $\psi(\boldsymbol{x}, \boldsymbol{y})$ is a formula in GF such that $\text{free}(\psi) \subseteq \text{free}(\alpha) = \boldsymbol{x} \cup \boldsymbol{y}$, then also the formulae

$$\exists \boldsymbol{y}(\alpha(\boldsymbol{x}, \boldsymbol{y}) \wedge \psi(\boldsymbol{x}, \boldsymbol{y})) \quad \text{and} \quad \forall \boldsymbol{y}(\alpha(\boldsymbol{x}, \boldsymbol{y}) \rightarrow \psi(\boldsymbol{x}, \boldsymbol{y}))$$

belong to GF.

Here $\text{free}(\psi)$ means the set of free variables of ψ. An atom $\alpha(\boldsymbol{x}, \boldsymbol{y})$ that relativizes a quantifier as in rule (3) is the *guard* of the quantifier. Hence in GF, guards must be atoms. But the really crucial property of guards (also for the more powerful guarded logics that we will consider below) is that it must contain *all* free variables of the formula that is quantified over.

The main motivation for introducing the guarded fragment was to explain and generalize the good algorithmic and model-theoretic properties of propositional *modal* logics (see [1, 26]). Recall that the basic (poly)modal logic ML (also called K_n) extends propositional logic by the possibility to construct formulae $\langle a \rangle \psi$ and $[a]\psi$ (for any a from a given set A of 'actions' or 'modalities') with the meaning that ψ holds at some, respectively, each, a-successor of the current state. (We refer to [4] or [22] for background on modal logic).

Although ML is formally a propositional logic we really view it as a fragment of first-order logic. Kripke structures, which provide the semantics for modal logics, are just relational structures with only unary and binary relations. There is a standard translation taking every formula $\psi \in \text{ML}$ to a first-order formula $\psi^*(x)$ with one free variable, such that for every Kripke structure K with a distinguished node w we have that $K, w \models \psi$ if and only if $K \models \psi^*(w)$. This translation takes an atomic proposition P to the atom Px, it commutes with the Boolean connectives, and it translates the modal operators by quantifiers as follows:

$$\langle a \rangle \psi \rightsquigarrow (\langle a \rangle \psi)^*(x) := \exists y (E_a xy \wedge \psi^*(y))$$
$$[a]\psi \rightsquigarrow ([a]\psi)^*(x) := \forall y (E_a xy \rightarrow \psi^*(y))$$

where E_a is the transition relation associated with the modality a. The *modal fragment* of first-order logic is the image of propositional modal logic under this translation. Clearly the translation of modal logic into first-order logic uses only guarded quantification, so we see immediately that the modal fragment is contained in GF. The guarded fragment generalizes the modal fragment by dropping the restrictions to use only two variables and only monadic and binary predicates, and retains only the restriction that quantifiers must be guarded.

The following properties of GF have been demonstrated [1, 10]:

(1) The satisfiability problem for GF is decidable.
(2) GF has the finite model property, i.e., every satisfiable formula in the guarded fragment has a finite model.
(3) GF has (a generalized variant of) the tree model property.
(4) Many important model theoretic properties which hold for first-order logic and modal logic, but not, say, for the bounded-variable fragments FO^k, do hold also for the guarded fragment.
(5) The notion of equivalence under guarded formulae can be characterized by a straightforward generalization of bisimulation.

Further work on the guarded fragment can be found in [7–9, 17]. Based on this kind of results Andréka, van Benthem, and Németi put forward the 'thesis' that it is the guarded nature of quantification that is the main reason for the good model-theoretic and algorithmic properties of modal logics.

Let us discuss to what extent this explanation is adequate. One way to address this question is to look at the complexity of GF. We have shown in [10] that the satisfiability problem for GF is complete for 2EXPTIME, the class of problems solvable by a deterministic algorithm in time $2^{2^{p(n)}}$, for some polynomial $p(n)$. This seems very bad, in particular if we compare it to the well-known fact that the satisfiability problem for propositional modal logic is in PSPACE [19]. But dismissing the explanation of Andréka, van Benthem, and Németi on these grounds would be too superficial. Indeed, the reason for the double exponential time complexity of GF is just the fact that predicates may have unbounded arity (wheras ML only expresses properties of graphs). Given that even a single predicate of arity n over a domain of just two element leads to 2^{2^n} possible types already on the atomic level, the double exponential lower complexity bound is hardly a surprise. Further, in most of the potential applications of guarded logics the arity of the relation symbols is bounded. But for GF-sentences of bounded arity, the satisfiability problem can be decided in EXPTIME [10], which is a complexity level that is reached already for rather weak extensions of ML (e.g. by a universal modality) [25]. Thus, the complexity analysis does not really provide a decisive answer to our question.

To approach the question from a different angle, let us look at extensions of ML. Indeed ML is a very weak logic and the really interesting modal logics extend ML by features like path quantification, temporal operators, least and greatest fixed points etc. which are of crucial importance for most computer science applications. It has turned out that many of these extended modal logics are algorithmically still manageable and actually of considerable practical importance. The most important of these extensions is the modal μ-calculus L_μ, which extends ML by least and greatest fixed points and subsumes most of the modal logics used for automatic verification including CTL, LTL, CTL*, PDL, and also many description logics. The satisfiability problem for L_μ

is known to be decidable and complete for EXPTIME [5]. Therefore, a good test for the explanation put forward by Andréka, van Benthem, and Németi is the following problem:

> If we extend GF by least and greatest fixed points, do we still get a decidable logic? If yes, what is its complexity? To put it differently, what is the penalty, in terms of complexity, that we pay for adding fixed points to the guarded fragment?

In [13] we were able to give a positive answer to this question. The model-theoretic and algorithmic methods that are available for the μ-calculus on one side, and the guarded fragments of first-order logic on the other side, can indeed be combined and generalized to provide positive results for guarded fixed point logics. (Precise definitions for these logics will be given in the next section.) In fact we could establish precise complexity bounds.

Theorem 1.2 (Grädel, Walukiewicz). *The satisfiability problems for guarded fixed point logics are decidable and* 2EXPTIME*-complete. For guarded fixed point sentences of bounded width the satisfiability problem is* EXPTIME*-complete.*

By the width of a formula ψ, we mean the maximal number of free variables in the subformulae of ψ. For sentences that are guarded in the sense of GF, the width is bounded by the maximal arity of the relation symbols, but there are other variants of guarded logics where the width may be larger. Note that for guarded fixed point sentences of bounded width the complexity level is the same as for μ-calculus and for GF without fixed points.

The proof that we give in [13] relies on alternating two-way tree automata (on trees of unbounded branching), on a forgetful determinacy theorem for parity games, and on a notion of *tableaux* for guarded fixed point sentences, which can be viewed as tree representations of structures. We associate with every guarded fixed point sentence ψ an alternating tree automaton A_ψ that accepts precisely the tableaux that represent models for ψ. This reduces the satisfiability problem for guarded fixed point logic to the emptiness problem for alternating two-way tree automata.

In this paper we discuss other variants of guarded logics, with more liberal notions of guarded quantification and explain alternative possibilities to design decision procedures for guarded fixed point logics. Already in [3], van Benthem had proposed *loosely guarded quantification* (also called pairwise guarded quantification) as a more general way of restricting quantifiers, and proved that also LGF, the loosely guarded fragment of first-order logic, remains decidable. Here we motivate and introduce *clique guarded quantification*, which is even more liberal than loosely guarded quantification, but retains the same decidability properties.

The techniques for establishing decidability results for guarded fixed point logics that we explain in this paper exploit a crucial property of such logics, namely the (generalized variant of the) *tree model property*, saying that every satisfiable sentence of width k has a model of tree width at most $k-1$. The tree width of a structure is a notion coming from graph theory which measures how closely the structure resembles a tree. Informally a structure has tree width $\leqslant k$, if it can be covered by (possibly overlapping) substructures of size at most $k+1$ which are arranged in a tree-like manner. The tree model property for guarded logics is a consequence of their invariance under a suitable notion of bisimulation, called guarded bisimulation.

Guarded bisimulations play a fundamental role for characterizing the expressive power of guarded logics, in the same way as usual bisimulations are crucial for understanding modal logics. For instance, the characterization theorem by van Benthem [2], saying that a property is definable in propositional modal logic if and only if it is first-order definable and invariant under bisimulation, has a natural analogue for the guarded fragment: GF can define precisely the model classes that are first-order definable and invariant under guarded bisimulation [1]. We will explain and prove a similar result for the clique-guarded fragment in Section 3. There is a similar and highly non-trivial characterisation theorem for the modal μ-calculus, due to Janin and Walukiewicz [18], saying that the properties definable in the modal μ-calculus are precisely the properties that are definable in monadic second-order logic and invariant under bisimulation. And, as shown recently by Grädel et al. [12], this result also carries over to the guarded world. Indeed, there is a natural fragment of second-order logic, called GSO, which is between monadic second-order logic and full second-order logic, such that guarded fixed point logic is precisely the bisimulation-invariant portion of GSO.

1.1. Outline of the paper

In Section 2 we discuss different variants of guarded logics. In particular we introduce the notion of clique-guarded quantification and present the precise definitions and elementary properties of guarded fixed point logics. In Section 3 we explain the notions of guarded bisimulations, of tree width and of the unraveling of a structure. We prove a characterization theorem for the clique-guarded fragment and discuss the tree model property of guarded logics. Based on the tree model property we will present in Section 4.2 a simpler decidability proof for guarded fixed point logic that replaces the automata theoretic machinery used in [13] by an interpretation argument into the monadic second-order theory of countable trees (SωS) which by Rabin's famous result [23] is known to be decidable. We then show in Section 4.3 that instead of using SωS, one can also reduce guarded fixed point logic to the μ-calculus with backwards modalities which has recently been proved to be decidable (in Exptime, actually) by Vardi [27].

We remark that this paper is to a considerable extent expository. The new results, mostly concerning clique-guarded logics, can also be derived using the automata theoretic techniques of [13]. However, it is worthwhile to make the role of guarded bisimulations explicit and to show how one can establish decidability results for guarded fixed point logics via reductions to well-known formalisms such as $S\omega S$ or the μ-calculus. Even if the automata theoretic method gives more efficient algorithms, the reduction technique provides a simple and high-level method for proving decidability, avoiding any explicit use of automata-theoretic machinery (the use of automata is hidden in the decision algorithms for $S\omega S$ or the μ-calculus). For the convenience of the reader, we have included explicit proofs of some facts that are known – or straightforward variations of known results – but where the proofs are hard to find.

2. Guarded logics

There are several ways to define more general guarded logics than GF. On one side, we can consider other notions of guardedness, and on the other side we can look at guarded fragments of more powerful logics than first-order logic. We first consider other guardedness conditions.

2.1. Loosely guarded quantification

The direct translation of temporal formulae (ψ **until** φ), say over the temporal frame $(\mathbb{N}, <)$, into first-order logic is

$$\exists y(x \leqslant y \wedge \varphi(y) \wedge \forall z((x \leqslant z \wedge z < y) \rightarrow \psi(z))$$

which is not guarded in the sense of Definition 1.1. However, the quantifier $\forall z$ in this formula is guarded in a weaker sense, which lead van Benthem [3] to the following generalization of GF.

Definition 2.1. The *loosely guarded fragment* LGF is defined in the same way as GF, but the quantifier-rule is relaxed as follows:

(3)′ If $\psi(\boldsymbol{x},\boldsymbol{y})$ is in LGF, and $\alpha(\boldsymbol{x},\boldsymbol{y}) = \alpha_1 \wedge \cdots \wedge \alpha_m$ is a conjunction of atoms, then

$$\exists \boldsymbol{y}((\alpha_1 \wedge \cdots \wedge \alpha_m) \wedge \psi(\boldsymbol{x},\boldsymbol{y})) \quad \text{and} \quad \forall \boldsymbol{y}((\alpha_1 \wedge \cdots \wedge \alpha_m) \rightarrow \psi(\boldsymbol{x},\boldsymbol{y}))$$

belong to LGF, provided that $\text{free}(\psi) \subseteq \text{free}(\alpha) = \boldsymbol{x} \cup \boldsymbol{y}$ and for any two variables $z \in \boldsymbol{y}$, $z' \in \boldsymbol{x} \cup \boldsymbol{y}$ there is at least one atom α_j that contains both z and z'.

In the translation of (ψ **until** φ) described above, the quantifier $\forall z$ is loosely guarded by $(x \leqslant z \wedge z < y)$ since z coexists with both x and y in some conjunct of the guard. On the other side, the transitivity axiom $\forall xyz(Exy \wedge Eyz \rightarrow Exz)$ is not in LGF. The conjunction $Exy \wedge Eyz$ is not a proper guard of $\forall xyz$ since x and z do not coexist in any conjunct. Indeed, it has been shown in [10] that there is no way to express transitivity in LGF.

2.2. Clique-guarded quantification

In this paper we introduce a new, even more liberal, variant of guarded quantification, which leads to what we may call *clique-guarded logics*. To motivate this notion, let us look at the semantic meaning of guardedness.

Definition 2.2. Let $\mathfrak{B}$ be a structure with universe B and vocabulary τ. A set $X = \{b_1, \dots, b_n\} \subseteq B$ is *guarded* in $\mathfrak{B}$ if there exists an atomic formula $\alpha(x_1, \dots, x_n)$ such that $\mathfrak{B} \models \alpha(b_1, \dots, b_n)$. Note that every singleton set $X = \{b\}$ is guarded (by the atom $b = b$). A tuple $(b_1, \dots, b_n) \in B^n$ is guarded if $\{b_1, \dots, b_n\} \subseteq X$ for some guarded set X.

Clearly, sentences of GF can refer only to guarded tuples. Consider now the LGF-sentence

$$(\exists xyz\,.\,Rxyz)(\forall uv\,.\,Ryuv \wedge Rzuv)\varphi(y, z, u, v).$$

While the first quantifier is guarded even in the sense of GF, the second one is only loosely guarded: the quantified variables u, v coexist in an atom of the guard (in fact in both of them) and they also coexist with each of the other free variables y, z in one of the atoms. The subformula φ can hence talk about quadruples (y, z, u, v) in a structure, that are not guarded in the sense of the definition just given, but in weaker sense.

The corresponding semantic definition of a *loosely guarded* set in a structure $\mathfrak{B}$ is inductive.

Definition 2.3. A set X is loosely guarded in the structure $\mathfrak{B}$ if it either is a guarded set, or if there exists a loosely guarded set X' such that for every $a \in X - X'$ and every $b \in X$ there is a guarded set Y with $\{a, b\} \subseteq Y \subseteq X$.

For instance, in the structure $\mathfrak{B} = (B, R)$, with universe $B = \{a, b, c, d, e\}$ and a ternary relation consisting of the triples $(a, b, c), (b, d, e), (c, d, e)$, the set $\{b, c, d, e\}$ is loosely guarded. Note that the elements of a loosely guarded set need not coexist in a single atom of the structure, but they are all 'adjacent' in the sense of the locality graph or Gaifman graph of a structure.

Definition 2.4. The *Gaifman graph* of a relational structure $\mathfrak{B}$ (with universe B) is the undirected graph $G(\mathfrak{B}) = (B, E^{\mathfrak{B}})$ where

$$E^{\mathfrak{B}} = \{(a, a')\colon\ a \neq a',\ \text{there exists a guarded set } X \subseteq B \text{ with } a, a' \in X\}.$$

A set X of elements of a structure $\mathfrak{B}$ is *clique-guarded* in $\mathfrak{B}$ if it induces a clique in $G(\mathfrak{B})$. A tuple $(b_1, \dots, b_n) \in B^n$ is clique-guarded if its components form a clique-guarded set.

Lemma 2.5. *Every loosely guarded set is also clique-guarded.*

Proof. We proceed by induction on the definition of a loosely guarded set. If X is guarded, then it is obviously also clique-guarded. Otherwise there exists a loosely guarded set X' and, for every $a \in X - X'$, $b \in X$ a guarded set containing both a and b. Hence all such a and b are connected in $G(\mathfrak{B})$. It remains to consider elements

a, b that are both contained in $X \cup X'$. In that case, a and b are connected in $G(\mathfrak{B})$ because, by induction hypothesis, X' induces a clique in $G(\mathfrak{B})$. □

The converse is not true, as the following example shows. Consider a structure $\mathfrak{A} = (A, R)$ with universe $A = \{a_1, a_2, a_3, b_{12}, b_{23}, b_{13}\}$ and one ternary relation R containing the triangles $(a_1, a_2, b_{12}), (a_2, a_3, b_{23}), (a_1, a_3, b_{13})$. Then $\{a_1, a_2, a_3\}$ is neither guarded nor loosely guarded, but induces a clique in $G(\mathfrak{A})$.

Note that for each finite vocabulary τ and each $k \in \mathbb{N}$, there is a positive, existential first-order formula $clique(x_1, \ldots, x_k)$ such that, for every τ-structure $\mathfrak{B}$ and all $b_1, \ldots, b_k \in B$

$$\mathfrak{B} \models clique(b_1, \ldots, b_k) \Leftrightarrow b_1, \ldots, b_k \text{ induce a clique in } G(\mathfrak{B}).$$

Definition 2.6. The *clique-guarded fragment* CGF of first-order logic is defined in the same way as GF and LGF, but with the *clique*-formulae as guards. Hence, the quantification rule for CGF is

(3)″ If $\psi(\boldsymbol{x}, \boldsymbol{y})$ is a formula CGF, then

$$\exists \boldsymbol{y}(clique(\boldsymbol{x}, \boldsymbol{y}) \wedge \psi(\boldsymbol{x}, \boldsymbol{y})) \quad \text{and} \quad \forall \boldsymbol{y}(clique(\boldsymbol{x}, \boldsymbol{y}) \rightarrow \psi(\boldsymbol{x}, \boldsymbol{y}))$$

belong to CGF, provided that $\text{free}(\psi) \subseteq \text{free}(clique) = \boldsymbol{x} \cup \boldsymbol{y}$.

Note that quantifiers over tuples are in principle no longer needed in CGF (contrary to GF and LGF), since they can be written as sequences of clique-guarded quantifiers over single variables.

2.3. Alternative definitions of CGF

In practice, one will of course not write down the clique-formulae explicitly. One possibility is not to write them down at all, i.e., to take the usual (unguarded) first-order syntax and to change the semantics of quantifiers so that only clique-guarded tuples are considered. More precisely: let $\varphi(\boldsymbol{x}, y)$ be a first-order formula, $\mathfrak{B}$ a structure and $\boldsymbol{a}$ be a clique-guarded tuple of elements of $\mathfrak{B}$. Then $\mathfrak{B} \models \exists y \varphi(\boldsymbol{a}, y)$ if there exists an element b such that $(\boldsymbol{a}, b)$ is clique-guarded and $\mathfrak{B} \models \varphi(\boldsymbol{a}, b)$. Similarly for universal quantifiers. It is easy to see that, for finite vocabularies, this semantic definition of CGF is equivalent to the one given above.

An alternative possibility is to permit as guards any existential positive formula $\gamma(\boldsymbol{x})$ that implies $clique(\boldsymbol{x})$. This is what Maarten Marx [20, 16] uses in his definition of the *packed fragment* PF. The differences between the clique-guarded fragment and the packed fragment are purely syntactical. PF and CGF have the same expressive power. The work of Maarten Marx and ours have been done independently.

Every LGF-sentence is equivalent to a CGF-sentence. (The analogous statement for formulae is only true if we impose that the free variables must be interpreted by loosely guarded tuples. However, in this paper, we restrict attention to sentences.) We observe that CGF has strictly more expressive power than LGF.

Proposition 2.7. *The CGF-sentence* $\forall xyz(clique(x,y,z) \rightarrow Rxyz)$ *is not equivalent to any sentence in LGF.*

The good algorithmic and model-theoretic properties of GF go through also for LGF and CGF. In most cases, in particular for decidability, for the characterization via an appropriate notion of guarded bisimulation and for the tree model property, the proofs for GF extend without major difficulties. An exception is the finite model property which, for GF, has been established in [10], and where the extension to LGF and CGF, recently established by Ian Hodkinson [15], requires considerable effort.

Notation. We use the notation $(\exists \boldsymbol{y} \,.\, \alpha)$ and $(\forall \boldsymbol{y} \,.\, \alpha)$ for relativized quantifiers,

i.e., we write guarded formulae in the form $(\exists \boldsymbol{y} \,.\, \alpha)\psi(\boldsymbol{x},\boldsymbol{y})$ and $(\forall \boldsymbol{y} \,.\, \alpha)\psi(\boldsymbol{x},\boldsymbol{y})$. When this notation is used, then it is always understood that α is indeed a proper guard as specified by condition (3), $(3)'$, or $(3)''$.

2.4. Guarded fixed point logics

We now define guarded fixed point logics, which can be seen as the natural common extensions of GF, LGF and CGF on one side, and the μ-calculus on the other side.

Definition 2.8. The guarded fixed point logics μGF, μLGF, and μCGF are obtained by adding to GF, LGF, and CGF, respectively, the following rules for constructing fixed point formulae: Let W be a k-ary relation symbol, $\boldsymbol{x} = x_1,\dots,x_k$ a k-tuple of distinct variables, and $\psi(W,\boldsymbol{x})$ be a guarded formula that contains only positive occurrences of W, no free first-order variables other than $x_1,\dots,x_k$, and where W is not used in guards. Then we can build the formulae

$$[\text{LFP } W\boldsymbol{x} \,.\, \psi](\boldsymbol{x}) \quad \text{and} \quad [\text{GFP } W\boldsymbol{x} \,.\, \psi](\boldsymbol{x}).$$

The semantics of the fixed point formulae is the usual one: Given a structure $\mathfrak{A}$ providing interpretations for all free second-order variables in ψ, except W, the formula $\psi(W,\boldsymbol{x})$ defines an operator on k-ary relations $W \subseteq A^k$, namely

$$\psi^{\mathfrak{A}} : W \mapsto \psi^{\mathfrak{A}}(W) := \{\boldsymbol{a} \in A^k : \mathfrak{A} \models \psi(W,\boldsymbol{a})\}.$$

Since W occurs only positively in ψ, this operator is monotone (i.e., $W \subseteq W'$ implies $\psi^{\mathfrak{A}}(W) \subseteq \psi^{\mathfrak{A}}(W')$) and therefore has a least fixed point LFP($\psi^{\mathfrak{A}}$) and a greatest fixed point GFP($\psi^{\mathfrak{A}}$). Now, the semantics of least fixed point formulae is defined by

$$\mathfrak{A} \models [\text{LFP } W\boldsymbol{x} \,.\, \psi(W,\boldsymbol{x})](\boldsymbol{a}) \quad \text{iff } \boldsymbol{a} \in \text{LFP}(\psi^{\mathfrak{A}})$$

and similarly for the greatest fixed points.

Least and greatest fixed point can be defined inductively. For a formula $\psi(W, \boldsymbol{x})$ with k-ary relation variable W a structure $\mathfrak{A}$, and ordinals α set

$$\begin{aligned} W^0 &:= \emptyset, & \tilde{W}^0 &:= A^k, \\ W^{\alpha+1} &:= \psi^{\mathfrak{A}}(W^\alpha), & \tilde{W}^{\alpha+1} &:= \psi^{\mathfrak{A}}(\tilde{W}^\alpha), \\ W^\beta &:= \bigcup_{\alpha<\beta} W^\alpha, & \tilde{W}^\beta &:= \bigcap_{\alpha<\beta} \tilde{W}^\alpha \quad \text{for limit ordinals } \beta. \end{aligned}$$

The relations W^α (resp. $\tilde{W}^\alpha$) are called the *stages* of the LFP-induction (resp. GFP-induction) of $\psi(W, \boldsymbol{x})$ on $\mathfrak{A}$. Since the operator $\psi^{\mathfrak{A}}$ is monotone, we have $W^0 \subseteq W^1 \subseteq \cdots \subseteq W^\alpha \subseteq W^{\alpha+1} \subseteq \cdots$ and $\tilde{W}^0 \supseteq \tilde{W}^1 \supseteq \cdots \supseteq \tilde{W}^\alpha \supseteq \tilde{W}^{\alpha+1} \supseteq \cdots$, and there exist ordinals α, α' such that $W^\alpha = \text{LFP}(\psi^{\mathfrak{A}})$ and $\tilde{W}^{\alpha'} = \text{GFP}(\psi^{\mathfrak{A}})$. These are called the *closure ordinals* of the LFP-induction resp. GFP-induction of $\psi(W, \boldsymbol{x})$ on $\mathfrak{A}$.

2.5. Finite and countable models

Contrary to GF, LGF, CGF, and also to the modal μ-calculus, guarded fixed point logics do not have the finite model property [13].

Proposition 2.9. *Guarded fixed point logic μGF* (*even with only two variables*, *without nested fixed points and without equality*) *contains infinity axioms.*

Proof. Consider the conjunction of the formulae

$$\exists xyFxy$$

$$(\forall xy \,.\, Fxy)\exists xFyx$$

$$(\forall xy \,.\, Fxy)[\text{LFP}\, Wx \,.\, (\forall y \,.\, Fyx)Wy](x).$$

The first two formulae say that a model should contain an infinite F-path and the third formula says that F is well-founded, thus, in particular, acyclic. Therefore every model of the three formulae is infinite. On the other side, the formulae are clearly satisfiable, for instance by $(\omega, <)$. □

While the finite model property fails for guarded fixed point logics we recall, for future use, that the Löwenheim–Skolem property is true even for the (unguarded) least fixed point logic (FO + LFP), i.e., every satisfiable fixed point sentence has a countable model. This result is part of the folklore on fixed point logic, but it is hard to find a published proof. Our exposition follows the one in [6].

Theorem 2.10. *Every satisfiable sentence in* (FO + LFP), *and hence every satisfiable sentence in μGF, μLGF, or μCGF, has a model of countable cardinality.*

Proof. Consider a fixed point formula of form $\psi(\boldsymbol{x}) := [\text{LFP}\ R\boldsymbol{x} \,.\, \varphi(R, \boldsymbol{x})](\boldsymbol{x})$, with first-order φ such that $\mathfrak{A} \models \psi(\boldsymbol{a})$ for some infinite model $\mathfrak{A}$.

For any ordinal α, let R^α be stage α of the least fixed point induction of φ on $\mathfrak{A}$.

Expand $\mathfrak{A}$ by a monadic relation U, a binary relation $<$, and a $m+1$-ary relation S (where m is the arity of R) such that

(1) $(U,<)$ is a well-ordering of length $\gamma+1$, and $<$ is empty outside U.

(2) S describes the stages of $\varphi^{\mathfrak{A}}$ in the following way:

$$S := \{(u,\boldsymbol{b})\text{: for some ordinal } \alpha \leqslant \gamma,\ u \text{ is the } \alpha\text{th element of } (U,<), \text{ and } \boldsymbol{b} \in R^\alpha\}.$$

In the expanded structure $\mathfrak{A}^* := (\mathfrak{A}, U, <, S)$ the stages of the operator $\varphi^{\mathfrak{A}}$ are defined by the sentence

$$\eta := \forall u \forall \boldsymbol{x}(Su\boldsymbol{x} \leftrightarrow \exists z(z < u \wedge \varphi[R\boldsymbol{y}/\exists z(z < u \wedge Sz\boldsymbol{y})](\boldsymbol{x}))).$$

Here $\varphi[R\boldsymbol{y}/\exists z(z<u\wedge Sz\boldsymbol{y})](\boldsymbol{x}))$ is the formula obtained form $\varphi(R,\boldsymbol{x})$ by replacing all occurrences of subformula $R\boldsymbol{y}$ by $\exists z(z<u\wedge Sz\boldsymbol{y})$.

Let $\mathfrak{B}^* = (\mathfrak{B}, U^{\mathfrak{B}}, <^{\mathfrak{B}}, S^{\mathfrak{B}})$ be a countable elementary substructure of $\mathfrak{A}^*$, containing the tuple $\boldsymbol{a}$. Since $\mathfrak{A}^* \models \eta$, also $\mathfrak{B}^* \models \eta$ and therefore $S^{\mathfrak{B}}$ encodes the stages of $\varphi^{\mathfrak{B}}$. Since also $\mathfrak{B}^* \models \exists u S u \boldsymbol{a}$, it follows that $\boldsymbol{a}$ is contained in the least fixed point of $\varphi^{\mathfrak{B}}$, i.e., $\mathfrak{B} \models \psi(\boldsymbol{a})$.

A straightforward iteration of this argument gives the desired result for arbitrary nestings of fixed point operators, and hence for the entire fixed point logic FO + LFP. □

2.6. Guarded infinitary logics

It is well known that fixed point logics have a close relationship to infinitary logics (with bounded number of variables).

Definition 2.11. GF^∞, LGF^∞, and CGF^∞ are the infinitary variants of the guarded fragments GF, LGF, and CGF, respectively. For instance GF^∞ extends GF by the following rule for building new formulae: If $\Phi \subseteq \mathrm{GF}^\infty$ is any set of formulae, then also $\bigvee\Phi$ and $\bigwedge\Phi$ are formulae of GF^∞. The definitions for LGF^∞ and CGF^∞ are anologous.

In the sequel we explicitly talk about the clique-guarded case only, i.e., about μCGF and CGF^∞, but all results apply to the guarded and loosely guarded case as well. The following simple observation relates μCGF with CGF^∞.

Proposition 2.12. *For each $\psi \in \mu CGF$ of width k and each cardinal γ, there is a $\psi' \in CGF^\infty$, also of width k, which is equivalent to ψ on all structures up to cardinality γ.*

Proof. Consider a fixed point formula $[\mathrm{LFP}\ R\boldsymbol{x}\,.\,\varphi(R,\boldsymbol{x})](\boldsymbol{x})$. For every ordinal α, there is a formula $\varphi^\alpha(\boldsymbol{x}) \in \mathrm{CGF}^\infty$ that defines the stage α of the induction of φ. Indeed, let

$\varphi^0(\mathbf{x}) := \mathit{false}$, let $\varphi^{\alpha+1}(\mathbf{x}) := \varphi[R\mathbf{y}/\varphi^\alpha(\mathbf{y})](\mathbf{x})$, that is, the formula that one obtains from $\varphi(R, \mathbf{x})$ if one replaces each atom $R\mathbf{y}$ (for any $\mathbf{y}$) by the formula $\varphi^\alpha(\mathbf{y})$, and for limit ordinals β, let $\varphi^\beta(\mathbf{x}) := \bigvee_{\alpha<\beta} \varphi^\alpha(\mathbf{x})$. On structures of bounded cardinality, also the closure ordinal of any fixed-point formula is bounded. Hence for every cardinal γ there exists an ordinal α such that $[\mathrm{LFP}\ R\mathbf{x} \,.\, \varphi(R, \mathbf{x})](\mathbf{x})$ is equivalent to $\varphi^\alpha(\mathbf{x})$ on structures of cardinality at most γ. □

Remark. Without the restriction on the cardinality of the structures, this result fails. Indeed there are very simple fixed point formulae, even in the modal μ-calculus (such as well-foundedness axioms), that are not equivalent to any formulae of the full infinitary logic $L_{\infty\omega}$.

3. Guarded bisimulation and the tree model property

Tree width is an important notion in graph theory. Many difficult or undecidable computational problems on graphs become easy on graphs of bounded tree width. The tree width of a structure measures how closely it resembles a tree. Informally, a structure has tree width $\leqslant k$, if it can be covered by (possibly overlapping) substructures of size at most $k+1$ which are arranged in a tree-like manner. For instance, trees and forests have tree width 1, cycles have tree width 2, and the $(n \times n)$-grid has tree width n. Here we need the notion of tree width for arbitrary relational structures. For readers who are familiar with the notion of tree width in graph theory we can simply say that the tree width of a structure is the tree width of its Gaifman graph. Here is a more detailed definition.

Definition 3.1. A structure $\mathfrak{B}$ (with universe B) has *tree width* k if k is the minimal natural number satisfying the following condition. There exists a directed tree $T = (V, E)$ and a function $F : V \to \{X \subseteq B\colon |X| \leqslant k+1\}$, assigning to every node v of T a set $F(v)$ of at most $k+1$ elements of $\mathfrak{B}$, such that the following two conditions hold.

(i) For every guarded set X in $\mathfrak{B}$ there exists a node v of T with $X \subseteq F(v)$.

(ii) For every element b of $\mathfrak{B}$, the set of nodes $\{v \in V\colon b \in F(v)\}$ is connected (and hence induces a subtree of T).

For each node v of T, $F(v)$ induces a substructure $\mathfrak{F}(v) \subseteq \mathfrak{B}$ of cardinality at most $k+1$. (Since $F(v)$ may be empty, we also permit empty substructures.) $\langle T, (\mathfrak{F}(v)_{v \in T})\rangle$ is called a *tree decomposition* of width k of $\mathfrak{B}$.

Remark. A more concise, but equivalent, formulation of clause (i) would be that $\mathfrak{B} = \bigcup_{v \in T} \mathfrak{F}(v)$.

By definition, every guarded set $X \subseteq B$ is contained in some $F(v)$. A simple graph theoretic argument shows that the same is true for loosely guarded and clique-guarded sets.

Lemma 3.2. *Let $\langle T, (\mathfrak{F}(v)_{v\in T})\rangle$ be a tree decomposition of $\mathfrak{B}$ and $X \subseteq B$ be a clique-guarded set in $\mathfrak{B}$. Then there exists a node v of T such that $X \subseteq F(v)$.*

Proof. For each $b \in X$, let V_b be the set of nodes v such that $b \in F(v)$. By the definition of a tree decomposition, each V_b induces a subtree of T. For all $b, b' \in X$ the intersection $V_b \cap V_{b'}$ is non-empty, since b and b' are adjacent in $G(\mathfrak{B})$ and must therefore coexist in some atomic fact that is true in $\mathfrak{B}$. It is known that any collection of pairwise overlapping subtrees of a tree has a common node (see e.g. [24, p. 94]). Hence there is a node v of the T such that $F(v)$ contains all elements of X. □

3.1. Guarded bisimulations

The notion of bisimulation from modal logic generalises in a straightforward way to various notions of guarded bisimulation that describe indistinguishability in guarded logics. We focus here on *clique-bisimulations*, the appropriate notion for clique-guarded formulae. The notions of guarded or loosely guarded bisimulations can be defined analogously.

Definition 3.3. A *clique-k-bisimulation*, between two τ-structures $\mathfrak{A}$ and $\mathfrak{B}$ is a non-empty set I of finite partial isomorphisms $f : X \to Y$ from $\mathfrak{A}$ to $\mathfrak{B}$, where $X \subseteq A$ and $Y \subseteq B$ are clique-guarded sets of size at most k, such that the following back and forth conditions are satisfied. For every $f : X \to Y$ in I,

forth: for every clique-guarded set $X' \subseteq A$ of size at most k there exists a partial isomorphism $g : X' \to Y'$ in I such that f and g agree on $X \cap X'$.

back: for every clique-guarded set $Y' \subseteq B$ of size at most k there exists a partial isomorphism $g : X' \mapsto Y'$ in I such that f^{-1} and g^{-1} agree on $Y \cap Y'$.

Clique-bisimulations are defined in the same way, without restriction on the size of X, Y, X' and Y'. Two τ-structures $\mathfrak{A}$ and $\mathfrak{B}$ are *clique*-(k-)*bisimilar* if there exists a cliqe-(k-)bisimulation between them. Obviously, two structures are clique-bisimilar if and only if they are clique-k-bisimilar for all k.

Remark. One can describe clique-k-bisimilarity also via a guarded variant of the infinitary Ehrenfeucht–Fraïssé game with k pebbles. One just has to impose that after every move, the set of all pebbled elements induces a clique in the Gaifman graph of each of the two structures. Then $\mathfrak{A}$ and $\mathfrak{B}$ are clique-k-bisimilar if and only if Player II has a winning strategy for this guarded game.

Adapting basic and well-known model-theoretic techniques to the present situation, one obtains the following result.

Theorem 3.4. *Let $\mathfrak{A}$ and $\mathfrak{B}$ be two τ-structures. The following are equivalent*:

(i) *$\mathfrak{A}$ and $\mathfrak{B}$ are clique-k-bisimilar.*

(ii) *For all sentences $\psi \in CGF^{\infty}$ of width at most k, $\mathfrak{A} \models \psi \Leftrightarrow \mathfrak{B} \models \psi$.*

Proof. (i) $\Rightarrow$ (ii): Let I be a clique-k-bisimulation between $\mathfrak{A}$ and $\mathfrak{B}$, let $a_1,\dots,a_n \in A$, $b_1,\dots,b_n \in B$, and $\psi(x_1,\dots,x_n)$ be any formula in CGF$^\infty$ with width at most k such that $\mathfrak{A} \models \psi(a_1,\dots,a_n)$ and $\mathfrak{B} \models \neg\psi(b_1,\dots,b_n)$. We show, by induction on ψ, that there is no partial isomorphism $f \in I$ with $f(a_1)=b_1,\dots,f(a_n)=b_n$. By setting $n=0$, the claim follows.

If ψ is atomic this is obvious, and the induction steps for formulae $\psi = \bigvee\Phi$ and $\psi = \neg\varphi$ are immediate. Hence the only interesting case concerns formulae of the form

$$\psi(\boldsymbol{x}) := (\exists \boldsymbol{y} \,.\, clique(\boldsymbol{x},\boldsymbol{y}))\varphi(\boldsymbol{x},\boldsymbol{y}).$$

Since $\mathfrak{A} \models \psi(\boldsymbol{a})$, there exists a tuple $\boldsymbol{a}'$ in $\mathfrak{A}$ such that $\mathfrak{A} \models clique(\boldsymbol{a},\boldsymbol{a}') \wedge \varphi(\boldsymbol{a},\boldsymbol{a}')$. Suppose, towards a contradiction, that some $f \in I$ takes $\boldsymbol{a}$ to $\boldsymbol{b}$. Since the set $\boldsymbol{a} \cup \boldsymbol{a}'$ is clique-guarded there exists a partial isomorphism $g \in I$, taking $\boldsymbol{a}$ to $\boldsymbol{b}$ and $\boldsymbol{a}'$ to some tuple $\boldsymbol{b}'$ in $\mathfrak{B}$. But then the tuple $\boldsymbol{b} \cup \boldsymbol{b}'$ is clique-guarded and $\mathfrak{B} \models \neg\varphi(\boldsymbol{b},\boldsymbol{b}')$, which contradicts the induction hypothesis.

(ii) $\Rightarrow$ (i): Let I be the set of all partial isomorphisms $f : \boldsymbol{a} \mapsto \boldsymbol{b}$, taking a clique-guarded tuple $\boldsymbol{a}$ in $\mathfrak{A}$ to a clique-guarded tuple $\boldsymbol{b}$ in $\mathfrak{B}$ such that for all formulae $\psi(\boldsymbol{x}) \in$ CGF$^\infty$ of width at most k, $\mathfrak{A} \models \psi(\boldsymbol{a})$ iff $\mathfrak{B} \models \psi(\boldsymbol{b})$. Since $\mathfrak{A}$ and $\mathfrak{B}$ cannot be distinguished by sentences of width k in CGF$^\infty$, I contains the empty map and is therefore non-empty. It remains to show that I satisfies the back and forth properties.

For the forth property, take any partial isomorphism $f : X \to Y$ in I and any clique-guarded set X' in $\mathfrak{A}$ of size at most k. Let $X' = \{a_1,\dots,a_r,a'_1,\dots,a'_s\}$ where $X \cap X' = \{a_1,\dots,a_r\}$. We have to show that there exists a $g \in I$, defined on X' that coincides with f on $X \cap X'$.

Suppose that no such g exists. Let $\boldsymbol{a} = a_1,\dots,a_r$, $\boldsymbol{a}' = a'_1,\dots,a'_s$, $\boldsymbol{b} = f(\boldsymbol{a})$, and let T be the set of all tuples $\boldsymbol{b}' = b'_1,\dots,b'_s$ such that $\boldsymbol{b} \cup \boldsymbol{b}'$ is clique-guarded in $\mathfrak{B}$. Since there is no appropriate $g \in I$ there exists for every tuple $\boldsymbol{b}' \in T$ a formula $\varphi_{\boldsymbol{b}'}(\boldsymbol{x},\boldsymbol{y}) \in$ CGF$^\infty$ such that $\mathfrak{A} \models \varphi_{\boldsymbol{b}'}(\boldsymbol{a},\boldsymbol{a}')$ but $\mathfrak{B} \models \neg\varphi_{\boldsymbol{b}'}(\boldsymbol{b},\boldsymbol{b}')$. But then we can construct the formula

$$\psi(\boldsymbol{x}) := (\exists \boldsymbol{y} \,.\, clique(\boldsymbol{x},\boldsymbol{y})) \wedge \{\varphi_{\boldsymbol{b}'}(\boldsymbol{x},\boldsymbol{y}) \colon \boldsymbol{b}' \in T\}.$$

Clearly, $\mathfrak{A} \models \psi(\boldsymbol{a})$ but $\mathfrak{B} \models \neg\psi(\boldsymbol{b})$ which is impossible, given that $f \in I$ maps $\boldsymbol{a}$ to $\boldsymbol{b}$. The proof for the back property is analogous. □

In particular, this shows that clique-(k-)bisimilar structures cannot be separated by μCGF-sentences (of width k).

3.2. Characterizing CGF via clique-guarded bisimulations

We show next that the characterisations of propositional modal logic and GF as bisimulation-invariant fragments of first-order logic [1, 2] have their counterpart for CGF and clique-guarded bisimulation. The proof is a straightforward adaptation of van Benthems proof for modal logic, but for the convenience of the reader, we present it in full. However, we assume that the reader is familiar with the notions of elementary extensions and ω-saturated structures (see any textbook on model theory, such as [14, 21]). We recall that every structure has an ω-saturated elementary extension.

Theorem 3.5. *A first-order sentence is invariant under clique-guarded bisimulation if and only if it is equivalent to a CGF-sentence.*

Proof. We have already established that CGF-sentences (in fact even sentences from CGF^∞) are invariant under clique-guarded bisimulations. For the converse, suppose that ψ is a satisfiable first-order sentence that is invariant under clique-guarded bisimulations. Let Φ be the set of sentences $\varphi \in \mathrm{CGF}$ such that $\psi \models \varphi$. It suffices to show that $\Phi \models \psi$. Indeed, by the compactness theorem, already a finite conjunction of sentences from Φ will then imply, and hence be equivalent to, ψ.

Since ψ was assumed to be satisfiable, so is Φ. Take any model $\mathfrak{A} \models \Phi$. We have to prove that $\mathfrak{A} \models \psi$. Let $T_{\mathrm{CGF}}(\mathfrak{A})$ be the CGF-theory of $\mathfrak{A}$, i.e., the set of all CGF-sentences that hold in $\mathfrak{A}$.

Claim. *$T_{\mathrm{CGF}}(\mathfrak{A}) \cup \{\psi\}$ is satisfiable.*

Otherwise there were sentences $\varphi_1, \ldots, \varphi_n \in T_{\mathrm{CGF}}(\mathfrak{A})$ such that $\psi \models \neg(\varphi_1 \wedge \cdots \wedge \varphi_n)$. Hence $\neg(\varphi_1 \wedge \cdots \wedge \varphi_n)$ is a CGF-sentence implied by ψ and is therefore contained in Φ. But then $\mathfrak{A} \models \neg(\varphi_1 \wedge \cdots \wedge \varphi_n)$ which is impossible since $\varphi_1, \ldots, \varphi_n \in T_{\mathrm{CGF}}(\mathfrak{A})$. This proves the claim.

Take any model $\mathfrak{B} \models T_{\mathrm{CGF}}(\mathfrak{A}) \cup \{\psi\}$, and let $\mathfrak{A}^+$ and $\mathfrak{B}^+$ be ω-saturated elementary extensions of $\mathfrak{A}$ and $\mathfrak{B}$, respectively.

Claim. *$\mathfrak{A}^+$ and $\mathfrak{B}^+$ are clique-bisimilar.*

Let I be the set of partial isomorphisms $f : X \to Y$ from clique-guarded subsets of $\mathfrak{A}^+$ to clique-guarded subsets of $\mathfrak{B}^+$ such that, for all formulae $\varphi(\boldsymbol{x})$ in CGF and all tuples $\boldsymbol{a}$ from X, we have that $\mathfrak{A}^+ \models \varphi(\boldsymbol{a})$ iff $\mathfrak{B}^+ \models \varphi(f\boldsymbol{a})$. The fact that $\mathfrak{A}^+$ and $\mathfrak{B}^+$ are ω-saturated implies that the back and forth conditions for clique-guarded bisimulations are satisfied by I. Indeed, let $f \in X$, and let X' be any clique-guarded set in $\mathfrak{A}^+$, with $X' \cap X = \{a_1, \ldots, a_r\}$ and $X' - X = \{a'_1, \ldots, a'_s\}$. Let Φ be the set of all formulae of form $\varphi(fa_1, \ldots, fa_r, y_1, \ldots, y_s) \in \mathrm{CGF}$ such that $\mathfrak{A}^+ \models \varphi(a_1, \ldots, a_r, a'_1, \ldots, a'_s)$. For every formula $\varphi(f\boldsymbol{a}, \boldsymbol{y}) \in \Phi$, we have $\mathfrak{A}^+ \models (\exists \boldsymbol{y} \,.\, \mathit{clique}(\boldsymbol{a}, \boldsymbol{y}))\varphi(\boldsymbol{a}, \boldsymbol{y})$ and therefore also $\mathfrak{B}^+ \models (\exists \boldsymbol{y} \,.\, \mathit{clique}(f\boldsymbol{a}, \boldsymbol{y}))\varphi(f\boldsymbol{a}, \boldsymbol{y})$. Hence Φ is a consistent type of $(\mathfrak{B}^+, f\boldsymbol{a})$ which is, by ω-saturation, realized in $\mathfrak{B}^+$ by some fixed tuple $\boldsymbol{b}$ such that $(f\boldsymbol{a}, \boldsymbol{b})$ is clique-guarded. Hence the function g taking $\boldsymbol{a}$ to $f\boldsymbol{a}$ and $\boldsymbol{a}'$ to $\boldsymbol{b}$ is a partial isomorphism with domain X' that coincides with f on $X \cap X'$. The back property is proved in the same way, exploiting that $\mathfrak{A}^+$ is ω-saturated.

We can now complete the proof of the theorem. Since $\mathfrak{B} \models \psi$ and $\mathfrak{B}^+$ is an elementary extension of $\mathfrak{B}$, we have that $\mathfrak{B}^+ \models \psi$. By assumption, ψ is invariant under clique-guarded bisimulations, so $\mathfrak{A}^+ \models \psi$ and therefore also $\mathfrak{A} \models \psi$. □

An analogous result applies to clique-k-bisimulations and CGF-sentences of width k, for any $k \in \mathbb{N}$.

3.3. Unravelings of structures

The k-unraveling $\mathfrak{B}^{(k)}$ of a structure $\mathfrak{B}$ is defined inductively. We build a tree T, together with functions F and G such that for each node v of T, $F(v)$ induces a clique-guarded substructure $\mathfrak{F}(v) \subseteq \mathfrak{B}$, and $G(v)$ induces a substructure $\mathfrak{G}(v) \subseteq \mathfrak{B}^{(k)}$ that is isomorphic to $\mathfrak{F}(v)$. Further, $\langle T, (\mathfrak{G}(v))_{v \in T} \rangle$ will be a tree decomposition of $\mathfrak{B}^{(k)}$.

The root of T is λ, with $F(\lambda) = G(\lambda) = \emptyset$. Given a node v of T with $F(v) = \{a_1, \ldots, a_r\}$ and $G(v) = \{a_1^*, \ldots, a_r^*\}$ we create for every clique-guarded set $\{b_1, \ldots, b_s\}$ in $\mathfrak{B}$ with $s \leqslant k$ a successor node w of v such that $F(w) = \{b_1, \ldots, b_s\}$ and $G(w)$ is a set $\{b_1^*, \ldots, b_s^*\}$ which is defined as follows. For those i, such that $b_i = a_j \in F(v)$, put $b_i^* = a_j^*$ so that $G(w)$ has the same overlap with $G(v)$ as $F(w)$ has with $F(v)$. The other b_i^* in $G(w)$ are fresh elements.

Let $f_w : F(w) \to G(w)$ be the bijection taking b_i to b_i^* for $i = 1, \ldots, s$. For $\mathfrak{F}(w)$ being the substructure of $\mathfrak{B}$ induced by $F(w)$, define $\mathfrak{G}(w)$ so that f_w is an isomorphism from $\mathfrak{F}(w)$ to $\mathfrak{G}(w)$. Finally $\mathfrak{B}^{(k)}$ is the structure with tree decomposition $\langle T, (\mathfrak{G}(v)_{v \in T}) \rangle$.

Note that the k-unraveling of a structure has tree width at most $k - 1$.

Proposition 3.6. *$\mathfrak{B}$ and $\mathfrak{B}^{(k)}$ are k-bisimilar.*

Proof. Let I be the set of functions $f_v : F(v) \to G(v)$ for all nodes v of T. □

It follows that no sentence of width k in CGF^{∞}, and hence no sentence of width k in μCGF distinguishes between a structure and its k-unraveling. Since every satisfiable sentence in μCGF has a model of at most countable cardinality, and since the k-unraveling of a countable model is again countable we obtain the following tree model property for guarded fixed point logic.

Theorem 3.7 (Tree model property). *Every satisfiable sentence in μCGF with width k has a countable model of tree width at most $k - 1$.*

Remark. In fact the decision algorithms for guarded fixed point logics imply a stronger version of the tree model property, where the underlying tree has branching bounded by $\mathrm{O}(|\psi|^k)$ (see [13]).

4. Decision procedures

Once the tree model property is established, there are several ways to design decision algorithms for guarded logics. We focus here on guarded fixed point logics (in fact on μCGF which contains μGF and μLGF).

4.1. Tree representations of structures

Let $\langle T, (\mathfrak{F}(v))_{v\in T}\rangle$ be a tree decomposition of width $k-1$ of a τ-structure $\mathfrak{D}$ with universe D. We want to describe $\mathfrak{D}$ by a tree with a finite set of labels. To this end, we fix a set K of $2k$ constants and choose a function $f: D \to K$ assigning to each element d of $\mathfrak{D}$ a constant $a_d \in K$ such that the following condition is satisfied. If v, w are adjacent nodes of T, then distinct elements of $\mathfrak{F}(v) \cup \mathfrak{F}(w)$ are always mapped to distinct constants of K.

For each constant $a \in K$, let O_a be the set of those nodes $v \in T$ at which the constant a occurs, i.e., for which there exists an element $d \in \mathfrak{F}(v)$ such that $f(d) = a$. Further, we introduce for each m-ary relation R of $\mathfrak{D}$ a tuple $\bar{R} := (R_{\boldsymbol{a}})_{\boldsymbol{a}\in K^m}$ of monadic relations on T with

$$
\begin{aligned}
R_{\boldsymbol{a}} := \{v \in T\colon & \text{ there exist } d_1,\dots,d_m \in \mathfrak{F}(v) \text{ such that} \\
& \mathfrak{F}(v) \models Rd_1\cdots d_m \text{ and } f(d_1)=a_1,\dots,f(d_m)=a_m\}.
\end{aligned}
$$

The tree $T = (V, E)$ together with the monadic relations O_a and $R_{\boldsymbol{a}}$ (for $R \in \tau$) is called the tree structure $T(\mathfrak{D})$ associated with $\mathfrak{D}$ (and, strictly speaking, with its tree decomposition and with K and f).

Lemma 4.1. *Two occurrences of a constant $a \in K$ at nodes u, v of T represent the same element of $\mathfrak{D}$ if and only if a occurs in the label of* all *nodes on the link between u and v. (The link between two nodes u, v in a tree T is the smallest connected subgraph of T containing both u and v.)*

An arbitrary tree $T = (V, E)$ with monadic relations O_a and $\bar{R}$ does define a tree decomposition of width $k-1$ of some structure $\mathfrak{D}$, provided that the following axioms are satisfied.

(1) At each node v, at most k of the predicates O_a are true.
(2) Neighbouring nodes agree on their common elements. For all m-ary relation symbols $R \in \tau$ we have the axiom

$$
\mathit{consistent}(\bar{R}) := \bigwedge_{\boldsymbol{a}\in K^m} \forall x \forall y \left(\left(Exy \wedge \bigwedge_{a\in \boldsymbol{a}} (O_a x \wedge O_a y) \right) \rightarrow (R_{\boldsymbol{a}} x \leftrightarrow R_{\boldsymbol{a}} y) \right).
$$

These are first-order axioms over the vocabulary $\tau^* := \{E\} \cup \{O_a : a \in K\} \cup \{R_{\boldsymbol{a}} : R \in \tau, \boldsymbol{a} \subseteq K\}$. Given a tree structure T with underlying tree $T = (V, E)$ and monadic predicates O_a and $R_{\boldsymbol{a}}$ satisfying (1) and (2), we obtain a structure $\mathfrak{D}$ such that $T(\mathfrak{D}) = T$ as follows. For every constant $a \in K$, we call two nodes u, w of T a-equivalent if $T \models O_a v$ for all nodes v on the link between u and w. Clearly this is an equivalence relation on O_a^T. We write $[v]_a$ for the a-equivalence class of the node v. The universe of $\mathfrak{D}$ is the set of all a-equivalence classes of T for $a \in K$, i.e.,

$$
D := \{[v]_a\colon\ v \in T,\ a \in K,\ T \models O_a v\}.
$$

For every m-ary relation symbol R in τ, we define

$$R^{\mathfrak{D}} := \{([v_1]_{a_1},\ldots,[v_m]_{a_m}):\ T \models R_{a_1\ldots a_m}v \text{ for some (and hence all) } v \in [v_1]_{a_1} \cap \cdots \cap [v_m]_{a_m}\}.$$

4.2. Reduction to SωS

We now describe a translation from μCGF into monadic second-order logic on countable trees. Given a formula $\varphi(x_1,\ldots,x_m) \in \mu$CGF and a tuple $\boldsymbol{a} = a_1,\ldots,a_m$ over K, we construct a monadic second-order formula $\varphi_{\boldsymbol{a}}(z)$ with one free variable. The formulae $\varphi_{\boldsymbol{a}}(z)$ describe in the associated tree structure $T(\mathfrak{D})$ the same properties of clique-guarded tuples as $\varphi(\boldsymbol{x})$ does in $\mathfrak{D}$. (We will make this statement more precise below).

On a directed tree $T = (V,E)$ we can express that U contains all nodes on the link between x and y by the formula

$$connect(U,x,y) := Ux \wedge Uy \wedge \exists r(Ur \wedge \forall z(Ezr \rightarrow \neg Uz) \wedge \forall w \forall z(Ewz \wedge Uz \wedge z \neq r \rightarrow Uw)).$$

For any set $\boldsymbol{a} \subseteq K$ we can then construct a monadic second-order formula

$$link_{\boldsymbol{a}}(x,y) := \exists U \left(connect(U,x,y) \wedge \forall z \left(Uz \rightarrow \bigwedge_{a\in\boldsymbol{a}} O_a z \right)\right)$$

saying that the tuple $\boldsymbol{a}$ occurs at all nodes on the link between x and y. The translation is now defined by induction as follows:

(1) If $\varphi(\boldsymbol{x})$ is an atom $Sx_{i_1}\cdots x_{i_m}$ then $\varphi_{\boldsymbol{a}}(z) := S_{\boldsymbol{b}}z$ where $\boldsymbol{b} = (a_{i_1},\ldots,a_{i_m})$.
(2) If $\varphi = (x_i = x_j)$, let $\varphi_{\boldsymbol{a}}(z) = true$ if $a_i = a_j$ and $\varphi_{\boldsymbol{a}}(z) = false$ otherwise.
(3) If $\varphi(\boldsymbol{x}) := clique(\boldsymbol{x})$, let

$$clique_{\boldsymbol{a}}(z) := \bigwedge_{a,a\in\boldsymbol{a}} \exists y \left(link_{a,a'}(y,z) \wedge \bigvee_{R\in\tau} \bigvee_{\boldsymbol{b}:a,a'\in\boldsymbol{b}} R_{\boldsymbol{b}}y \right).$$

(4) If $\varphi = \eta \wedge \vartheta$, let $\varphi_{\boldsymbol{a}}(z) = \eta_{\boldsymbol{a}}(z) \wedge \vartheta_{\boldsymbol{a}}(z)$.
(5) If $\varphi = \neg\vartheta$, let $\varphi_{\boldsymbol{a}}(z) = \neg\vartheta_{\boldsymbol{a}}(z)$.
(6) If $\varphi = (\exists \boldsymbol{y} \,.\, clique(\boldsymbol{x},\boldsymbol{y}))\eta(\boldsymbol{x},\boldsymbol{y})$, let

$$\varphi_{\boldsymbol{a}}(z) := \exists y \left(link_{\boldsymbol{a}}(y,z) \wedge \bigvee_{\boldsymbol{b}} \left(\bigwedge_{b\in\boldsymbol{b}} O_b y \wedge clique_{\boldsymbol{ab}}(y) \wedge \eta_{\boldsymbol{ab}}(y) \right)\right).$$

(7) If $\varphi = [\text{LFP} \quad S\boldsymbol{x} \,.\, \eta(S,\boldsymbol{x})](\boldsymbol{x})$, let

$$\varphi_{\boldsymbol{a}}(z) := \forall \bar{S} \left(\left(consistent(\bar{S}) \wedge \bigwedge_{\boldsymbol{b}} \forall x (S_{\boldsymbol{b}}x \leftrightarrow \eta_{\boldsymbol{b}}(\bar{S},x)) \right) \rightarrow S_{\boldsymbol{a}}z \right).$$

Here $\bar{S}$ is a tuple $(S_{\boldsymbol{b}})_{\boldsymbol{b}\in K^m}$ of monadic predicates where m is the arity of S.

Theorem 4.2. *Let $\varphi(\boldsymbol{x})$ be a formula in μCGF and $\mathfrak{D}$ be a structure with tree decomposition $\langle T, (\mathfrak{F}(v))_{v\in T}\rangle$. For an appropriate set of constants K and a function $f: D \to K$, let $T(\mathfrak{D})$ be the associated tree structure. Then, for every node v of T and every clique-guarded tuple $\boldsymbol{d} \subseteq \mathfrak{F}(v)$ with $f(\boldsymbol{d}) = \boldsymbol{a}$,*

$$\mathfrak{D} \models \varphi(\boldsymbol{d}) \Leftrightarrow T(\mathfrak{D}) \models \varphi_{\boldsymbol{a}}(v).$$

Proof. We proceed by induction on φ. The non-trivial cases are the clique-guards, existential quantification and least fixed points.

For the clique-guards, note that the translated formula $clique_{\boldsymbol{a}}(v)$ says that for any pair a, a' of components of $\boldsymbol{a}$, there is a node w, such that

- a, a' occur at all nodes on the link from v to w and hence represent the same elements d, d' at w as they do at v.
- $T(\mathfrak{D}) \models R_{\boldsymbol{b}}w$ for some predicate R and some tuple $\boldsymbol{b}$ that contains both a and a'. By induction hypothesis, this means that d, d' are components of some tuple $\boldsymbol{d}'$ such that $\mathfrak{D} \models R\boldsymbol{d}'$.

Hence $T(\mathfrak{D}) \models clique_{\boldsymbol{a}}(v)$ if and only if the tuple $\boldsymbol{d}$ induces a clique in the Gaifman graph $G(\mathfrak{D})$.

Suppose now that $\varphi(\boldsymbol{x}) = (\exists \boldsymbol{y}\,.\,clique(\boldsymbol{x},\boldsymbol{y}))\eta(\boldsymbol{x},\boldsymbol{y})$ and that $\mathfrak{D} \models \varphi(\boldsymbol{d})$. Then there exists a tuple $\boldsymbol{d}'$ such that $\mathfrak{D} \models clique(\boldsymbol{d},\boldsymbol{d}') \wedge \eta(\boldsymbol{d},\boldsymbol{d}')$.

By Lemma 3.2 there exists a node w of T such that all components of $\boldsymbol{d} \cup \boldsymbol{d}'$ are contained in $\mathfrak{F}(w)$.

Let $f(\boldsymbol{d}') = \boldsymbol{b}$. By induction hypothesis it follows that

$$T(\mathfrak{D}) \models \bigwedge_{b\in \boldsymbol{b}} O_b w \wedge clique_{\boldsymbol{ab}}(w) \wedge \eta_{\boldsymbol{ab}}(w).$$

Let U be the set of nodes on the link between v and w. Then the tuple $\boldsymbol{d}$ occurs in $\mathfrak{F}(u)$ for all nodes $u \in U$. It follows that $T(\mathfrak{D}) \models link_{\boldsymbol{a}}(v,w)$. Hence $T(\mathfrak{D}) \models \varphi_{\boldsymbol{a}}(v)$.

Conversely, if $T(\mathfrak{D}) \models \varphi_{\boldsymbol{a}}(v)$ then there exists a node w such that the constants $\boldsymbol{a}$ occur at all nodes on the link between v and w (and hence correspond to the same tuple $\boldsymbol{d}$) and such that $T(\mathfrak{D}) \models clique_{\boldsymbol{ab}}(w) \wedge \eta_{\boldsymbol{ab}}(w)$ for some tuple $\boldsymbol{b}$. By induction hypothesis this implies that $\mathfrak{D} \models clique(\boldsymbol{d},\boldsymbol{d}') \wedge \eta(\boldsymbol{d},\boldsymbol{d}')$ for some tuple $\boldsymbol{d}'$, hence $\mathfrak{D} \models \varphi(\boldsymbol{d})$.

Finally, let $\varphi(\boldsymbol{x}) = [\text{LFP} \;\; S\boldsymbol{x}\,.\,\eta(S,\boldsymbol{x})](\boldsymbol{x})$. By definition, $\mathfrak{D} \models \varphi(\boldsymbol{d})$ is true if and only if $\boldsymbol{d}$ is contained in every fixed point of the operator $\eta^{\mathfrak{D}}$, i.e., is in every relation S such that $S = \{\boldsymbol{c}\colon\; (\mathfrak{D},S) \models \eta(S,\boldsymbol{c})\}$.

We first observe that, for guarded tuples $\boldsymbol{d}$, this is equivalent to the seemingly weaker condition that $\boldsymbol{d}$ is contained in every S such that $\boldsymbol{c} \in S$ iff $\mathfrak{D} \models \eta(S,\boldsymbol{c})$ *for all guarded tuples* $\boldsymbol{c}$. Indeed, this is obvious since $\eta(S,\boldsymbol{x})$ is a Boolean combination of quantifier-free formulae not involving $\boldsymbol{x}$, of positive atoms of the form $S\boldsymbol{u}$ where $\boldsymbol{u}$ is a recombination of the variables appearing in $\boldsymbol{x}$ and of formulae starting with a guarded existential quantifier. Therefore the truth values of $S\boldsymbol{c}$ for unguarded tuples $\boldsymbol{c}$ never matters for the question whether a given guarded tuple is in $\varphi^{\mathfrak{D}}(S)$.

Recall that the formula associated with $\varphi(\boldsymbol{x})$ and $\boldsymbol{a}$ is

$$\varphi_{\boldsymbol{a}}(z) := (\forall \bar{S}) \left(\left(consistent(\bar{S}) \wedge \bigwedge_{\boldsymbol{b}} \forall x (S_{\boldsymbol{b}} x \leftrightarrow \eta_{\boldsymbol{b}}(\bar{S}, x)) \right) \rightarrow S_{\boldsymbol{a}} z \right).$$

Consider any tuple $\bar{S} = (S_{\boldsymbol{b}})_{\boldsymbol{b} \in K^m}$ of monadic relations on $T(\mathfrak{D})$ that satisifies the consistency axiom such that

$$(T(\mathfrak{D}), \bar{S}) \models \bigwedge_{\boldsymbol{b}} \forall x (S_{\boldsymbol{b}} x \leftrightarrow \eta_{\boldsymbol{b}}(\bar{S}, x)).$$

This tuple $\bar{S}$ defines a relation S on $\mathfrak{D}$ such that for all nodes w of T and all tuples $\boldsymbol{c}$ in $\mathfrak{F}(w)$ with $f(\boldsymbol{c}) = \boldsymbol{b}$, we have $\boldsymbol{c} \in S$ iff $w \in S_{\boldsymbol{b}}$. Conversely, each relation S on $\mathfrak{D}$ defines such a tuple $\bar{S}$ of monadic relations on $T(\mathfrak{D})$ which describes the truth values of S on all guarded tuples of $\mathfrak{D}$. Since $T(\mathfrak{D}) \models S_{\boldsymbol{b}} w \leftrightarrow \eta_{\boldsymbol{b}}(\bar{S}, w)$ it follows by induction hypothesis that $\mathfrak{D} \models S\boldsymbol{c} \leftrightarrow \eta(S, \boldsymbol{c})$. Further $\boldsymbol{d} \in S$ if and only if $v \in S_{\boldsymbol{a}}$.

Hence the formula $\varphi_{\boldsymbol{a}}(v)$ is true in $T(\mathfrak{D})$ if and only if $\boldsymbol{d}$ is contained in all relations S over $\mathfrak{D}$ such that for all guarded tuples $\boldsymbol{c}$, $\boldsymbol{c} \in S$ iff $\boldsymbol{c} \in \eta^{\mathfrak{D}}(S)$. By the remarks above, this is equivalent to saying that $\boldsymbol{d}$ is in the least fixed point of $\eta^{\mathfrak{D}}$. □

Theorem 4.3. *The satisfiability problem for μCGF is decidable.*

Proof. Let ψ be a sentence in μCGF of vocabulary τ and width k. We translate ψ into a monadic second-order sentence ψ^* such that ψ is satisfiable if and only if there exists a countable tree $T = (V, E)$ with $T \models \psi^*$.

Fix a set K of $2k$ constants and let $\bar{O}$ be the tuple of monadic relations O_a for $a \in K$. Further, for each m-relation symbol $R \in \tau$, let $\bar{R}$ be the tuple of monadic relation $R_{\boldsymbol{a}}$ where $\boldsymbol{a} \in K^m$. The desired monadic second-order sentence has the form

$$\psi^* := (\exists \bar{O})(\exists \bar{R})(\chi \wedge \forall x \psi_{\emptyset}(x)).$$

Here χ is the first-order axiom expressing that the tree T expanded by the relations $\bar{O}$ and $\bar{R}$ does describe a tree structure $T(\mathfrak{D})$ associated to some τ-structure $\mathfrak{D}$. We have shown above that this can be done in first-order logic. The formula $\psi_{\emptyset}(x)$ is the translation of ψ (and the empty tuple of constants) into monadic second-order logic, as described by Theorem 4.2.

If ψ is satisfiable, then by Theorem 3.7, ψ has a countable model $\mathfrak{D}$ of tree width $k-1$. By Theorem 4.2, the associated tree structure $T(\mathfrak{D})$ satisfies $\chi \wedge \forall x \psi_{\emptyset}(x)$, hence there exists a tree T such that $T \models \psi^*$. Conversely, if $T \models \psi^*$, then there exists an expansion $T = (T, \bar{O}, \bar{R})$ which satisfies χ and hence describes the tree decomposition of a τ-structure $\mathfrak{D}$. Since $T \models \forall x \psi_{\emptyset}(x)$, it follows by Theorem 4.2 that $\mathfrak{D} \models \psi$.

The decidability of μCGF now follows by the decidability of SωS, the monadic second-order theory of countable trees, a famous result that has been established by Rabin [23]. □

Note that while this reduction argument to SωS gives a somewhat more elementary decidability proof (modulo Rabin's result, of course), it does not give good complexity bounds. Indeed, even the first-order theory of countable trees is non-elementary, i.e. its time complexity exceeds every bounded number of iterations of the exponential function.

4.3. Reduction to the μ-calculus with backwards modalities

Instead of reducing the satisfiability problem for μCGF to the monadic second-order theory of trees, we can define a similar reduction to the μ-calculus with backward modalities and then invoke Vardi's decidability result for this logic [27].

For a set of actions A, the μ-calculus with backwards modalities $L_\mu^{\leftarrow}$, permits, for each action $a \in A$, besides the common modal operators $\langle a \rangle$ and $[a]$ also the backwards operators $\langle a^{\leftarrow} \rangle$ and $[a^{\leftarrow}]$ corresponding to the backwards transitions $E_a^{\leftarrow} := \{(w,v): (v,w) \in E_a\}$. Hence $\langle a^{\leftarrow} \rangle \varphi$ is true at state w in a Kripke structure K if and only if there exists a state v, such that $K, v \models \varphi$ and w is reachable from v via action a.

Here we will need $L_\mu^{\leftarrow}$ on trees (V,E) with only one transition relation. We can write $\langle + \rangle$, $[+]$ for the forward modal operators, and $\langle - \rangle$, $[-]$ for the backwards operators, and then use the abbreviations

$$\Diamond \varphi := \langle + \rangle \varphi \vee \langle - \rangle \varphi \quad \text{and} \quad \Box \varphi := [+]\varphi \wedge [-]\varphi.$$

Hence $\Diamond$ and $\Box$ are the usual modal operators on symmetric Kripke structures.

Finally, it is convenient for our reduction argument to permit the use of simultaneous least and greatest fixed points in $L_\mu^{\leftarrow}$. Let $\bar{X} = (X_1, \ldots, X_r)$ be a sequence of propositional variables, and $\bar{\varphi}(\bar{X}) = (\varphi_1(\bar{X}), \ldots, \varphi_r(\bar{X}))$ be a sequence of $L_\mu^{\leftarrow}$-formulae in which all occurrences of $X_1, \ldots, X_r$ are positive. Then, for each $i \leqslant r$, the expressions $[\mu \bar{X} \,.\, \bar{\varphi}(\bar{X})]_i$ and $[\nu \bar{X} \,.\, \bar{\varphi}(\bar{X})]_i$ are formulae in $L_\mu^{\leftarrow}$.

On every Kripke structure K with universe V, the sequence $\bar{\varphi}(\bar{X})$ defines an operator $\bar{\varphi}^K$ that maps any tuple $\bar{S} = (S_1, \ldots, S_r)$ of subsets $S_i \subseteq V$ to a new tuple $(\varphi_1^K(\bar{S}), \ldots, \varphi_r^K(\bar{S}))$ where $\varphi_i^K(\bar{S}) = \{v : K, v \models \varphi_i(\bar{S})\}$.

Since the variables in $\bar{X}$ occur only positive in $\bar{\varphi}$, the operator $\bar{\varphi}^K$ has a least fixed point $\bar{X}^\infty = (X_1^\infty, \ldots, X_r^\infty)$. Now, the semantics of simultaneous least fixed point formulae is given by

$$K, v \models [\mu \bar{X} . \bar{\varphi}(\bar{X})]_i \Leftrightarrow v \in X_i^\infty.$$

The meaning of a simultaneous greatest fixed point $[\nu \bar{X} \,.\, \bar{\varphi}(\bar{X})]_i$ is defined similarly. It is well-known that simultaneous fixed points can be rewritten as nestings of simple fixed points, so the use of simultaneous fixed points does not change the expressive power of $L_\mu^{\leftarrow}$.

Theorem 4.4 (Vardi). *Every satisfiable formula in $L_\mu^{\leftarrow}$ has a tree model. Further, the satisfiability problem for $L_\mu^{\leftarrow}$ is decidable and* Exptime*-complete.*

On connected Kripke structures (in particular on trees), the universal modality is definable in $L_\mu^{\leftarrow}$. For every formula φ, we write $\forall\varphi$ to abbreviate the formula $\mu X.\varphi \wedge \Box X$. It is easy to see that $\forall\varphi$ is satisfied at some state of a connected Kripke structure K if and only if φ is satisfied at *all* states of K.

Let $\mathfrak{D}$ be a τ structure of bounded tree width, and let $T(\mathfrak{D})$ be its tree representation as described in Section 4.1. We view $T(\mathfrak{D})$ as a Kripke structure, with atomic propositions O_a and $R_{\boldsymbol{a}}$. Having available an universal modality, the axioms for tree representations $T(\mathfrak{D})$ given in the previous subsection, can easily be expressed by modal formulae. For instance, the consistency axioms can be written

$$\mathit{consistent}(\bar{R}) := \forall \bigwedge_{\boldsymbol{a}\in K^m} \left(\bigwedge_{a\in\boldsymbol{a}} O_a \wedge R_{\boldsymbol{a}} \rightarrow \Box \left(\bigvee_{a\in\boldsymbol{a}} \neg O_a \vee R_{\boldsymbol{a}} \right) \right).$$

Theorem 4.5. *Let $\mathfrak{D}$ be a structure with tree decomposition $\langle T, (\mathfrak{F}(v))_{v\in T}\rangle$. For an appropriate set of constants K and a function $f: D \rightarrow K$, let $T(\mathfrak{D})$ be the associated tree structure. For every formula $\varphi(x_1,\dots,x_m)$ in μCGF and every tuple $\boldsymbol{a}\in K^m$ we can construct a formula $\varphi_{\boldsymbol{a}} \in L_\mu^{\leftarrow}$ such that, for every node v of T and every clique-guarded tuple $\boldsymbol{d} \subseteq \mathfrak{F}(v)$ with $f(\boldsymbol{d}) = \boldsymbol{a}$,*

$$\mathfrak{D} \models \varphi(\boldsymbol{d}) \Leftrightarrow T(\mathfrak{D}), \quad v \models \varphi_{\boldsymbol{a}}.$$

Proof. The translation is very similar to the translation into monadic second-order logic that was given in the previous section.

(1) If $\varphi(\boldsymbol{x})$ is an atom $Sx_{i_1}\cdots x_{i_m}$ then $\varphi_{\boldsymbol{a}} := S_{\boldsymbol{b}}$ where $\boldsymbol{b} = (a_{i_1},\dots,a_{i_m})$.
(2) If $\varphi = (x_i = x_j)$, then $\varphi_{\boldsymbol{a}} := \mathit{true}$ if $a_i = a_j$ and $\varphi_{\boldsymbol{a}} := \mathit{false}$ otherwise.
(3) For the guard formulae $\mathit{clique}(\boldsymbol{x})$, let

$$\mathit{clique}_{\boldsymbol{a}} := \bigwedge_{a,a'\in\boldsymbol{a}} \mu X. \left(\bigvee_{\substack{R\in\tau \\ \boldsymbol{b}\,:\, a,a'\in\boldsymbol{b}}} R_{\boldsymbol{b}} \vee \Diamond(O_a \wedge O_{a'} \wedge X) \right).$$

(4) If $\varphi = \eta \wedge \vartheta$, then $\varphi_{\boldsymbol{a}} := \eta_{\boldsymbol{a}} \wedge \vartheta_{\boldsymbol{a}}$.
(5) If $\varphi = \neg\vartheta$, then $\varphi_{\boldsymbol{a}} := \neg\vartheta_{\boldsymbol{a}}$.
(6) If $\varphi = (\exists \boldsymbol{y}\,.\,\mathit{clique}(\boldsymbol{x},\boldsymbol{y}))\eta(\boldsymbol{x},\boldsymbol{y})$, then

$$\varphi_{\boldsymbol{a}} := \bigvee_{\boldsymbol{b}} \mu X. \left(\left(\bigwedge_{b\in\boldsymbol{b}} O_b \wedge \mathit{clique}_{\boldsymbol{ab}} \wedge \eta_{\boldsymbol{ab}} \right) \vee \Diamond \left(\bigwedge_{a\in\boldsymbol{a}} O_a \wedge X \right) \right).$$

(7) If $\varphi := [\mathrm{LFP}\quad S\boldsymbol{x}\,.\,\eta(S,\boldsymbol{x})](\boldsymbol{x})$, let

$$\varphi_{\boldsymbol{a}} := [\mu\bar{S}\,.\,\bar{\eta}(\bar{S}))]_{\boldsymbol{a}}.$$

Here $\bar{S}$ is a tuple of fixed point varriables $s_{\boldsymbol{b}}$ and $\bar{\eta}(\bar{S})$ is the tuple of the formulae $\eta_{\boldsymbol{b}}(\bar{S})$ for all $\boldsymbol{b}\in K^m$ (where m is the arity of S).

The proof that the translation is correct is analogous to the proof of Theorem 4.2. $\Box$

We now get another proof for the decidability of guarded fixed point logic. Given a sentence $\psi \in \mu$CGF, we translate it into the $L_\mu^{\leftarrow}$-sentence $\psi_\emptyset$ according to Theorem 4.5 and take the conjunction with the consistency axioms in $L_\mu^{\leftarrow}$ for tree representations $T(\mathfrak{D})$. Then use Vardi's decidability result for $L_\mu^{\leftarrow}$. By the tree model property of $L_\mu^{\leftarrow}$, the tree model property of μCGF and Theorem 4.5 this gives a decision procedure for μCGF.

However, it is not clear whether this argument can be modified to provide the optimal complexity bounds for guarded fixed point logic.

References

[1] H. Andréka, J. van Benthem, I. Németi, Modal languages and bounded fragments of predicate logic, J. Philos. Logic 27 (1998) 217–274.
[2] J. van Benthem, Modal correspondence theory, Ph.D. Thesis, University of Amsterdam, 1976.
[3] J. van Benthem, Dynamic bits and pieces, ILLC Research Report, University of Amsterdam, 1997.
[4] P. Blackburn, M. de Rijke, Y. Venema, Modal Logic, Cambridge University Press, Cambridge, 2001.
[5] A. Emerson, C. Jutla, The complexity of tree automata and logics of programs, in: Proc. 29th IEEE Symp. on Foundations of Computer Science, 1988, pp. 328–337.
[6] J. Flum, On the (infinite) model theory of fixed-point logics, in: X. Caicedo, C. Montenegro (Eds.), Models, Algebras, and Proofs, Lecture Notes in Pure and Applied Mathematics Series, Vol. 2003, Marcel Dekker, New York, 1999, pp. 67–75.
[7] H. Ganzinger, H. de Nivelle, A superposition decision procedure for the guarded fragment with equality, in: Proc. 14th IEEE Symp. on Logic in Computer Science, 1999, pp. 295–305.
[8] H. Ganzinger, C. Meyer, M. Veanes, The two-variable guarded fragment with transitive relations, in: Proc. 14th IEEE Symp. on Logic in Computer Science, 1999, pp. 24–34.
[9] G. Gottlob, E. Grädel, H. Veith, Datalog lite: a deductive query language with linear time model checking, ACM Trans. Comput. Logic, to appear.
[10] E. Grädel, On the restraining power of guards, J. Symbolic Logic 64 (1999) 1719–1742.
[11] E. Grädel, Why are modal logics so robustly decidable?, Bull. European Assoc. Theoret. Comput. Sci. 68 (1999) 90–103.
[12] E. Grädel, C. Hirsch, M. Otto, Back and forth between guarded and modal logics, in: Proc. 15th IEEE Symp. on Logic in Computer Science, 2000, pp. 217–228.
[13] E. Grädel, I. Walukiewicz, Guarded fixed point logic, in: Proc. 14th IEEE Symp. on Logic in Computer Science, 1999, pp. 45–54.
[14] W. Hodges, Model Theory, Cambridge University Press, Cambridge, 1993.
[15] I. Hodkinson, Loosely guarded fragment of first-order logic has the finite model property, preprint, 2000.
[16] E. Hoogland, M. Marx, Interpolation in guarded fragments, preprint.
[17] E. Hoogland, M. Marx, M. Otto, Beth definability for the guraded fragment, in: Proc. LPAR'99, Lecture Notes in Artificial Intelligence, Springer, Berlin, 1999.
[18] D. Janin, I. Walukiewicz, On the expressive completeness of the propositional mu-calculus with respect to monadic second order logic, in: Proc. 7th Internat. Conf. on Concurrency Theory CONCUR'96, Lecture Notes in Computer Science, Vol. 1119, Springer, Berlin, 1996, pp. 263–277.
[19] R. Ladner, The computational complexity of provability in systems of propositional modal logic, SIAM J. Comput. 6 (1977) 467–480.
[20] M. Marx, Tolerance logic, Tech. Rep. IR-469, Faculteit der Exacte Wetenschappen, Vrije Universiteit Amsterdam, 1999.
[21] B. Poizat, Cours de théorie des modèles, Nur al-Mantiq wal-Ma'rifah, 1985.
[22] S. Popcorn, First Steps in Modal Logic, Cambridge University Press, Cambridge, 1994.
[23] M. Rabin, Decidability of second-order theories and automata on infinite trees, Trans. AMS 141 (1969) 1–35.

[24] B. Reed, Tree width and tangles: a new connectivity measure and some applications, in: R. Bailey (Ed.), Surveys in Combinatorics, Cambridge University Press, Cambridge, 1997, pp. 87–162.
[25] E. Spaan, Complexity of modal logics, Ph.D. Thesis, University of Amsterdam, Institute for Logic, Language and Computation, 1993.
[26] M. Vardi, Why is modal logic so robustly decidable?, in: N. Immerman, P. Kolaitis (Eds.), Descriptive Complexity and Finite Models, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 31, AMS, Providence, RI, 1997, pp. 149–184.
[27] M. Vardi, Reasoning about the past with two-way automata, in: Automata, Languages and Programming ICALP 98, Lecture Notes in Computer Science, Vol. 1443, Springer, Berlin, 1998, pp. 628–641.