Decision procedures for guarded logics

Erich Grädel

Mathematische Grundlagen der Informatik, RWTH Aachen,
graedel@informatik.rwth-aachen.de

Abstract. Different variants of guarded logics (a powerful generalization of modal logics) are surveyed and the recent decidability result for guarded fixed point logic (obtained in joint work with I. Walukiewicz) is explained. 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, different methods for reasoning with guarded fixed point sentences are presented:

(1) reduction to the monadic theory of countable trees (SωS);
(2) reduction to the μ-calculus with backwards modalities;
(3) the automata theoretic method (which gives theoretically optimal complexity results).

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 refer only to tuples of 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

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

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 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, van Benthem and Nemeti [1].

Definition 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. 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. 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 $\psi \in$ ML to a first-order formula $\psi^*(x)$ with one free variable, such that for every Kripke structure $\mathcal{K}$ with a distinguished node w we have that $\mathcal{K}, w \models \psi$ if and only if $\mathcal{K} \models \psi^*(w)$. 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. In fact the guarded fragment can be viewed as the generalization of the modal fragment of FO that is obtained by dropping the restrictions to use only two variables and only monadic and binary predicates, and to retain only the restriction that quantifiers must be guarded.

The following properties of GF have been demonstrated [1, 5]:

(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.

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 responsible 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 [5] that the the satisfiability problems 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. 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 vocabulary is fixed and the arity therefore bounded. But for GF-sentences of bounded arity, the satisfiability problem can be decided in EXPTIME [5], which is a complexity level that is reached already for rather weak extensions of ML (e.g. by a universal modality). 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 extension is the modal μ-calculus L_μ, which extends ML by least and greeatest 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 [3]. 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 [6] 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 can establish precise complexity bounds.

Theorem 1 (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 [6] 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. A tableau can be viewed as a tree representation of a structure. We associate with every guarded fixed point sentence ψ an alternating tree automaton $\mathcal{A}_\psi$ 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 explain this result and discuss several ways to design decision procedures for guarded fixed point logics. All these techniques exploit one very important property of guarded logics, namely the (generalized variant of the) *tree model property*, which says that every satisfiable sentence of width k has a model of tree width at most $k-1$.

In Sect.2 we discuss different variants of guarded logics. In Sect. 3 we explain the notions of guarded bisimulations and of the unraveling of a structure, and we prove the tree model property of guarded logics. Based on the tree model property we will present in Sect. 4.2 a somewhat simpler decidability proof for guarded fixed point logic that replaces the automata theoretic machinery used in [6] by an interpretation argument into the monadic second-order theory of countable trees (SωS) which by Rabin's famous result [8] is known to be decidable. We then show in Sect. 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 proved to be decidable (in EXPTIME actually) by Vardi [12]. Finally we sketch the direct automata theoretic approach used in [6] (but here without the explicit use of tableaux).

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.

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

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

which is not guarded in the sense of Definition 1. However, the quantifier $\forall z$ in this formula is guarded in a weaker sense, which lead van Benthem [2] to the following generalization of GF.

Definition 2. 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 $(x \leq z \wedge z < y)$ 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 [5] that there is no way to express transitivity in LGF.

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 3. Let $\mathfrak{B}$ be a structure with universe B and vocabulary τ. A set $X \subseteq B$ is *strictly k-guarded* in $\mathfrak{B}$ if, for some $s \leq k$, there exists $b_1, \ldots, b_s \in B$ such that

(i) $X \subseteq \{b_1, \ldots, b_s\}$, and
(ii) there exists an atom $\alpha(b_1, \ldots, b_s)$, in which all of $b_1, \ldots, b_s$ occur, such that $\mathfrak{B} \models \alpha(b_1, \ldots, b_s)$.

To put it differently, X is strictly k-guarded if there is a guard α in the sense of GF that guards a set containing X with cardinality at most k.

Similarly, a set $X \subseteq B$ is *loosely k-guarded* if it is contained in a set $\{b_1, \ldots, b_s\}$ (with $s \leq k$) that is guarded in the sense of LGF. That is clause *(ii)* is replaced by *(ii)'* There exists a conjunction $\alpha(b_1, \dots, b_s)$ of atoms such that every pair b_i, b_j coexists in some conjunct of α and $\mathfrak{B} \models \alpha(b_1, \dots, b_s)$.

A set is (strictly or loosely) guarded if it is (strictly or loosely) k-guarded for some k, and a tuple $\boldsymbol{b} = (b_1, \dots, b_r)$ is (strictly or loosely) guarded in $\mathfrak{B}$ if the set $\{b_1, \dots, b_r\}$ is.

Clearly, sentences of GF resp. LGF can refer only to strictly resp. loosely guarded tuples. Note that the components of a loosely guarded tuple need not coexist in a single atom, but they are nevertheless all 'adjacent' in the sense of the locality graph or Gaifman graph of a structure, which is an important notion in finite model theory. Let us recall the definition.

Definition 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

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

The following observation is obvious.

Lemma 1. *Let X be loosely guarded in $\mathfrak{B}$. Then 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 $(a_1, a_1, b_{12}), (a_2, a_3, b_{23})(a_1, a_3, b_{13})$. Then the tuple (a_1, a_2, a_3) is neither guarded nor loosely guarded, but induces a clique in $G(\mathfrak{A})$.

Definition 5. For each finite vocabulary τ and each $k \in \mathbb{N}$, there is a positive, existential first-order formula $\mathit{clique}(x_1, \dots, x_k)$ such that, for every τ-structure $\mathfrak{B}$ and all $b_1, \dots, b_k \in B$

$$\mathfrak{B} \models \mathit{clique}(b_1, \dots, b_k) \iff b_1, \dots, b_k \text{ induce a clique in } G(\mathfrak{B}).$$

A tuple $b_1, \dots, b_k$ such that $\mathfrak{B} \models \mathit{clique}(b_1, \dots, b_k)$ is called *clique-guarded* in $\mathfrak{B}$.

Definition 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

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

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

We observe that CGF has strictly more expressive power than LGF.

Proposition 1. *The* CGF*-sentence $\forall xyz(\mathit{clique}(x, y, z) \rightarrow Rxyz)$ is not equivalent to any sentence in* LGF.

On the other side CGF inherits all the nice decidability and model-theoretic properties of LGF.

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)$''$.

Remark. 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.

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 7. 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, $\boldsymbol{x} = x_1, \ldots, x_k$ 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 $x_1, \ldots, x_k$ and where W is not used in guards. Then we can build the formulae

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

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 $\text{LFP}(\psi^{\mathfrak{A}})$ and a greatest fixed point $\text{GFP}(\psi^{\mathfrak{A}})$. Now, the semantics of least fixed point formulae is defined by

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

and similarly for the greatest fixed points.

The Löwenheim-Skolem property. For future use, we recall that even the (unguarded) least fixed point logic (FO + LFP) has the Löwenheim-Skolem property. This result is part of the folklore on fixed point logic [4].

Theorem 2. *Every satisfiable sentence in* (FO + LFP)*, and hence every satisfiable sentence in* μGF*,* μLGF *or* μCGF*, has a model of countable cardinality.*

On the other side, guarded fixed point logics do *not* have the *finite model property* (see [6] for a simple counterexample).

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

Definition 8. $\mathrm{GF}^\infty, \mathrm{LGF}^\infty$ 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. *For each $\psi \in \mu\mathrm{CGF}$ of width k and each cardinal γ, there is a $\psi' \in \mathrm{CGF}^\infty$, also of width k, which is equivalent to ψ on all structures up to cardinality γ.*

3 Guarded bisimulation and the tree model property

Tree width is an important notion in graph theory. Here we need a generalisation of this concept to 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 9. A structure $\mathfrak{B}$ (with universe B and arbitrary vocabulary τ) 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

$$F : V \to \{X \subseteq B : |X| \leq k + 1\},$$

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 τ-atom $\alpha(x_1, \dots, x_r)$ and every tuple $b_1, \dots, b_r$ such that $\mathfrak{B} \models \alpha(b_1, \dots, b_r)$ there exists a node v of T such that $\{b_1, \dots, b_r\} \subseteq F(v)$.
(ii) For every element b of $\mathfrak{B}$, the set of nodes $\{v \in V : b \in F(v)\}$ is connected (and hence induces a subtree of T).

For each node v of T, the set $F(v)$ 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 strictly 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 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. [9, p. 94]). Hence there is a node v of the T such that $F(v)$ contains all elements of X. $\square$

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-k-bisimulations*, the appropriate notion for clique-guarded formulae of width at most k. The notions of strict or loose k-bisimulations can be defined analogously.

Definition 10. A *clique-k-bisimulation* (in the sequel simply called a 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 $Y \cap Y'$.

Two τ-structures $\mathfrak{A}$ and $\mathfrak{B}$ are *(clique-)k-bisimilar* if there exists a k-bisimulation between them.

Remark. One can describe 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 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. *Let $\mathfrak{A}$ and $\mathfrak{B}$ be two τ-structures. The following are equivalent:*

(i) $\mathfrak{A}$ and $\mathfrak{B}$ are k-bisimilar.
(ii) For all sentences $\psi \in \mathrm{CGF}^\infty$ of width at most k, $\mathfrak{A} \models \psi \iff \mathfrak{B} \models \psi$.

In particular this shows that k-bisimilar structures cannot be separated by guarded fixed point sentences of width k.

Unravelings of structures. The k-unraveling $\mathfrak{B}^{(k)}$ of a structure $\mathfrak{B}$ is defined inductively. We build a tree T, with functions F and G such that each

$F(v)$ induces a clique-guarded substructure $\mathfrak{F}(v) \subseteq \mathfrak{B}$, each $G(v)$ induces a substructure $\mathfrak{G}(v) \subseteq \mathfrak{B}^{(k)}$ that is isomorphic to $\mathfrak{F}(v)$, and $\langle T, (\mathfrak{G}(v))_{v\in T}\rangle$ is a tree decomposition of $\mathfrak{B}^{(k)}$.

The root of T is λ, with $F(\lambda) = G(\lambda) = \varnothing$. Given a node v of T with $F(v) = \{a_1, \dots, a_r\}$ and $G(v) = \{a_1^*, \dots, a_r^*\}$ we create for every clique-guarded set $\{b_1, \dots, b_s\}$ in $\mathfrak{B}$ with $s \leq k$ a successor node w of v such that $F(w) = \{b_1, \dots, b_s\}$ and $G(w)$ is a set $\{b_1^*, \dots, b_s^*\}$ 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 $i = 1, \dots, s$. 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. *$\mathfrak{B}$ 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 4 (Tree model property). *Every satisfiable sentence ψ in μCGF of width k has a countable model of tree width at most $k-1$.*

Remark. In fact our decision algorithms will imply a stronger version of the tree model property, where the underlying tree has branching bounded by $O(|\psi|^k)$.

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 $\mathcal{O}_a$ 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 $\overline{R} := (R_{\boldsymbol{a}})_{\boldsymbol{a} \in K^m}$ of monadic relations on T with

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

The tree $T = (V, E)$ together with the monadic relations $\mathcal{O}_a$ and $R_{\boldsymbol{a}}$ (for $R \in \tau$) is called the tree structure $\mathcal{T}(\mathfrak{D})$ associated with $\mathfrak{D}$ (and, strictly speaking, with its tree decomposition and with K and f). Note that 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 $\mathcal{O}_a$ and $\overline{R}$ 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 $\mathcal{O}_a$ are true.
(2) Neighbouring nodes agree on their common elements. For all m-ary relation symbols $R \in \tau$ we have the axiom

$$\mathit{consistent}(\overline{R}) := \bigwedge_{\boldsymbol{a} \in K^m} \forall x \forall y \Big(\Big(Exy \wedge \bigwedge_{a \in \boldsymbol{a}} (\mathcal{O}_a x \wedge \mathcal{O}_a y) \Big) \rightarrow (R_{\boldsymbol{a}} x \leftrightarrow R_{\boldsymbol{a}} y) \Big).$$

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

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

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

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

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 $\mathcal{T}(\mathfrak{D})$ 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

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

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

$$\mathit{link}_{\boldsymbol{a}}(x, y) := \exists U\Big(\mathit{connect}(U, x, y) \wedge \forall z(Uz \rightarrow \bigwedge_{a \in \boldsymbol{a}} \mathcal{O}_a z)\Big)$$

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 $\varphi_{\boldsymbol{a}}(z) = \mathit{true}$ if $a_i = a_j$ and $\varphi_{\boldsymbol{a}}(z) = \mathit{false}$ otherwise.
(3) If $\varphi(\boldsymbol{x}) := \mathit{clique}(\boldsymbol{x})$, let

$$\mathit{clique}_{\boldsymbol{a}}(z) := \bigwedge_{a, a \in \boldsymbol{a}} \exists y(\mathit{link}_{a,a'}(y, z) \wedge \bigvee_{R \in \tau} \bigvee_{\boldsymbol{b}: a, a' \in \boldsymbol{b}} R_{\boldsymbol{b}} y).$$

(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 $\varphi = (\exists \boldsymbol{y} \,.\, \mathit{clique}(\boldsymbol{x}, \boldsymbol{y}))\eta(\boldsymbol{x}, \boldsymbol{y})$, let

$$\varphi_{\boldsymbol{a}}(z) := \exists y\Big(\mathit{link}_{\boldsymbol{a}}(y, z) \wedge \bigvee_{\boldsymbol{b}}\Big(\bigwedge_{b \in \boldsymbol{b}} \mathcal{O}_b y \wedge \mathit{clique}_{\boldsymbol{ab}}(y) \wedge \eta_{\boldsymbol{ab}}(y)\Big)\Big).$$

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

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

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

Theorem 5. *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 $f : D \rightarrow K$, let $\mathcal{T}(\mathfrak{D})$ be the associated tree structure. Then, for every node v of T and every clique-guarded tuple $\boldsymbol{d} \subseteq \mathfrak{F}(v)$ with $f(\boldsymbol{d}) = \boldsymbol{a}$,*

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

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

For the clique-guards, note that the translated formula $\mathit{clique}_{\boldsymbol{a}}(v)$ 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 $\mathcal{T}(\mathfrak{D}) \models \mathit{clique}_{\boldsymbol{a}}(v)$ if and only if the tuple $\boldsymbol{d}$ induces a clique in in the Gaifman graph $G(\mathfrak{D})$.

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

By Lemma 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

$$\mathcal{T}(\mathfrak{D}) \models \bigwedge_{b \in \boldsymbol{b}} \mathcal{O}_b w \wedge \mathit{clique}_{\boldsymbol{ab}}(w) \wedge \eta_{\boldsymbol{ab}}(w).$$

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 $\mathcal{T}(\mathfrak{D}) \models \mathit{link}_{\boldsymbol{a}}(v, w)$. Hence $\mathcal{T}(\mathfrak{D}) \models \varphi_{\boldsymbol{a}}(v)$.

Conversely, if $\mathcal{T}(\mathfrak{D}) \models \varphi_{\boldsymbol{a}}(v)$ 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 $\mathcal{T}(\mathfrak{D}) \models \mathit{clique}_{\boldsymbol{ab}}(w) \wedge \eta_{\boldsymbol{ab}}(w)$ for some tuple $\boldsymbol{b}$. By induction hypothesis this implies that $\mathfrak{D} \models \mathit{clique}(\boldsymbol{d}, \boldsymbol{d}') \wedge \eta(\boldsymbol{d}, \boldsymbol{d}')$ for some tuple $\boldsymbol{d}'$, hence $\mathfrak{D} \models \varphi(\boldsymbol{d})$.

Finally, let $\varphi(\boldsymbol{x}) = [\mathrm{LFP} \;\; S\boldsymbol{x} \,.\, \eta(S, \boldsymbol{x})](\boldsymbol{x})$. 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 $S = \{\boldsymbol{c} : (\mathfrak{D}, S) \models \eta(S, \boldsymbol{c})\}$.

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

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

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

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

This tuple $\overline{S}$ 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 $\overline{S}$ of monadic relations on $\mathcal{T}(\mathfrak{D})$ which describes the truth values of S on all guarded tuples of $\mathfrak{D}$. Since $\mathcal{T}(\mathfrak{D}) \models S_{\boldsymbol{b}} w \leftrightarrow \eta_{\boldsymbol{b}}(\overline{S}, w)$ 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 $\mathcal{T}(\mathfrak{D})$ 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 6. *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 $\overline{\mathcal{O}}$ be the tuple of monadic relations $\mathcal{O}_a$ for $a \in K$. Further, for each m-relation symbol $R \in \tau$, let $\overline{R}$ be the tuple of monadic relation $R_{\boldsymbol{a}}$ where $\boldsymbol{a} \in K^m$. The desired monadic second-order sentence has the form

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

Here χ is the first-order axiom expressing that the tree T expanded by the relations $\overline{\mathcal{O}}$ and $\overline{R}$ does describe a tree structure $\mathcal{T}(\mathfrak{D})$ associated to some τ-structure $\mathfrak{D}$. We have shown above that this can be done in first-order logic. The formula $\psi_{\varnothing}(x)$ is the translation of ψ (and the empty tuple of constants) into monadic second-order logic, as described by Theorem 5.

If ψ is satisfiable, then by Theorem 4, ψ has a countable model $\mathfrak{D}$ of tree width $k-1$. By Theorem 5, the associated tree structure $\mathcal{T}(\mathfrak{D})$ satisfies $\chi \wedge \forall x \psi_{\varnothing}(x)$, hence there exists a tree T such that $T \models \psi^*$. Conversely, if $T \models \psi^*$, then there exists an expansion $\mathcal{T} = (T, \overline{\mathcal{O}}, \overline{R})$ which satisfies χ and hence describes the tree decomposition of a τ-structure $\mathfrak{D}$. Since $\mathcal{T} \models \forall x \psi_{\varnothing}(x)$ it follows by Theorem 5 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 [8]. □

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 [12].

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 $\mathfrak{K}$ if and only if there exists a state v, such that $\mathfrak{K}, v \models \varphi$ 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 $\overline{X} = (X_1, \dots, X_r)$ be a sequence of propositional variables, and $\overline{\varphi}(\overline{X}) = (\varphi_1(\overline{X}), \dots, \varphi_r(\overline{X}))$ be a sequence of $L_\mu^\leftarrow$-formulae in which all occurrences of $X_1, \dots, X_r$ are positive. Then, for each $i \le r$, the expressions $[\mu \overline{X} \,.\, \overline{\varphi}(\overline{X})]_i$ and $[\nu \overline{X} \,.\, \overline{\varphi}(\overline{X})]_i$ are formulae in $L_\mu^\leftarrow$.

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

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

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

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

Theorem 7 (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 $\mathfrak{K}$ if and only if φ is satisfied at *all* states of $\mathfrak{K}$.

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

$$\mathit{consistent}(\overline{R}) := \forall \bigwedge_{\boldsymbol{a}\in K^m} \Big(\bigwedge_{a\in\boldsymbol{a}} \mathcal{O}_a \wedge R_{\boldsymbol{a}} \rightarrow \Box\Big(\bigvee_{a\in\boldsymbol{a}} \neg\mathcal{O}_a \vee R_{\boldsymbol{a}} \Big)\Big).$$

Theorem 8. *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 $\mathcal{T}(\mathfrak{D})$ 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 $f(\boldsymbol{d}) = \boldsymbol{a}$,*

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

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

$$\mathit{clique}_{\boldsymbol{a}} := \bigwedge_{a,a'\in\boldsymbol{a}} \mu X\, .\, \Big(\bigvee_{\substack{R\in\tau \\ \boldsymbol{b}:\, a,a'\in\boldsymbol{b}}} R_{\boldsymbol{b}} \vee \Diamond(\mathcal{O}_a \wedge \mathcal{O}_{a'} \wedge X)\Big).$$

(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

$$\varphi_{\boldsymbol{a}} := \bigvee_{\boldsymbol{b}} \mu X\, .\, \Big(\Big(\bigwedge_{b\in\boldsymbol{b}} \mathcal{O}_b \wedge \mathit{clique}_{\boldsymbol{ab}} \wedge \eta_{\boldsymbol{ab}}\Big) \vee \Diamond\Big(\bigwedge_{a\in\boldsymbol{a}} \mathcal{O}_a \wedge X\Big)\Big).$$

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

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

Here $\overline{S}$ is a tuple of fixed point variables $S_{\boldsymbol{b}}$ and $\overline{\eta}(\overline{S})$ is the tuple of the formulae $\eta_{\boldsymbol{b}}(\overline{S})$ 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 5. □

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 $\psi_\varnothing$ according to Theorem 8 and take the conjunction with the consistency axioms in $L_\mu^{\leftarrow}$ for tree representations $\mathcal{T}(\mathfrak{D})$. 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 8 this gives a decision procedure for μCGF.

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

4.4 Alternating tree automata

Finally we sketch the automata theoretic method used in [6] for solving the satisfiability problem for guarded fixed point logic. The general idea is to associate with every sentence ψ an alternating automaton $\mathcal{B}_\psi$ that accepts precisely the (descriptions of) tree models of ψ. This reduces the satisfiability problem for ψ to the emptiness problem for $\mathcal{B}_\psi$, a problem that is solvable in exponential time with respect to the number of states of the automaton.

Automata based procedures are very important for the satisfiability testing and model checking of modal logics. For the μ-calculus and its relatives, *alternating tree automata* seem to be the right model to get optimal complexity results. In [12] Vardi used alternating *two-way* tree automata (A2A) to establish decidability and EXPTIME-completeness of the μ-calculus with backward modalities. His model of A2A works on trees of bounded branching where nodes have outgoing arcs labelled by indices $1, 2, \dots, k$. In our decidability proof for guarded fixed point logic in [6] we use a different variant of alternating automata which work on trees of arbitrary, finite or infinite, degree. The automata do not make use of the orientation of the edges in the tree; they may proceed to any neighbour of the current node, i.e. to the parent or to any child. A general forgetful determinacy theorem for parity games can be used to reduce the emptiness problem for our automata to the emptiness problem for Vardi's automata.

Definition 11. An alternating two-way automaton on trees is a tuple $\mathcal{A} = \langle Q_\exists, Q_\forall, \Sigma, q_0, \delta, \Omega\rangle$ where $Q = Q_\exists \cup Q_\forall$ is a finite set of states, partitioned into existential and universal states, Σ is an input alphabet and $q_0 \in Q$ is an initial state. The transition function has the form

$$\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q \cup \{\circ q : q \in Q\})$$

Intuitively $q' \in \delta(q, \sigma)$ means that a copy of the automaton should stay at the current node and assume a new state q'. If $\circ q' \in \delta(q, \sigma)$ then a copy of the automaton should proceed to some neighbour of the current node and assume state q'. The function $\Omega : Q \rightarrow \mathbb{N}$ specifies the acceptance condition of the automaton.

Let $\mathcal{T} = \langle T, L : T \rightarrow \Sigma\rangle$ be a tree whose nodes are labelled by symbols from Σ. The notion of a run of $\mathcal{A}$ on $\mathcal{T}$ and the acceptance condition can be defined via a game $\mathcal{G}(\mathcal{A}, \mathcal{T})$, played by two players, player 0 and player 1, who move a token along the edges of a graph.

Let $\mathcal{G}(\mathcal{A}, \mathcal{T}) = \langle V_0, V_1, E, \Omega\rangle$ where

- $V_0 = T \times Q_\exists$ and $V_1 = T \times Q_\forall$;
- $(v, q)E(v', q')$ iff either
  - *(i)* $q' \in \delta(q, L(v))$ and $v' = v$, or
  - *(ii)* $\circ q' \in \delta(q, L(v))$ and v' is a neighbour of v.
- $\Omega(v, q) = \Omega(q)$.

Initially, the token is at the vertex (λ, q_0) where λ is the root of T and q_0 is the initial state of the automaton. When the token is at a vertex $(v, q) \in V_j$, then player j moves it along some edge $(v, q)E(v', q')$ to a new node (v', q'). If one of the players cannot make a move she looses. Otherwise the result of the game is an infinite path $(\lambda, q_0), (v_1, q_1), (v_2, q_2), \ldots$ through the graph. In that case player 0 wins iff the sequence $\Omega(q_0), \Omega(q_1), \Omega(q_2), \ldots$ satisfies the parity condition: the minimal number $i \in \mathbb{N}$ appearing infinitely often in the sequence is even. (Therefore such games are called parity games.) By definition the automaton $\mathcal{A}$ accepts $\mathcal{T}$ if and only if player 0 has a winning strategy from position (λ, q_0) in $\mathcal{G}(\mathcal{A}, \mathcal{T})$.

Definition 12. A *memoryless strategy* for player 0 is a partial function $f : (T \times Q_\exists) \rightarrow T \times Q$ such that $(v, q)E(v', q')$ whenever $f(v, q) = (v', q')$. A finite or infinite play $(w_1, q_1), (w_2, q_2), \ldots$ of $\mathcal{G}(\mathcal{A}, \mathcal{T})$ is *consistent with* f if for every pair (w_i, q_i) such that $q_i \in Q_\exists$ the value of $f(w_i, q_i)$ is defined and $(w_{i+1}, q_{i+1}) = f(w_i, q_i)$. A memoryless strategy is *winning* from position (v, q) if player 0 wins every play starting at (v, q) that is consistent with f.

For a large class of games, there are *forgetful determinacy theorems* which say that from each position, one of the two players has a memoryless winning strategy. In particular this is the case for the graph games with parity winning conditions that we use here (see e.g. [13, Theorem 1]).

Theorem 9. *The automaton $\mathcal{A}$ accepts $\mathcal{T}$ if and only there exists a memoryless winning strategy for player 0 on $\mathcal{G}(\mathfrak{A}, \mathcal{T})$ from the initial position (λ, q_0).*

As a consequence, one can derive the following results (see [6] for further details).

Theorem 10. *If an alternating two-way automaton accepts some input tree, then it also accepts a tree whose branching is bounded by the number of states of the automaton. The emptiness problem for alternating two-way automata (on trees with arbitrary branching) can be decided in* EXPTIME.

Accepting tree models by automata. In [6] we construct for every guarded fixed point sentence ψ an automaton $\mathcal{A}_\psi$ accepting exactly those tableaux that represent a model for ψ. We do not explain the notion of a tableau here. Instead we present an almost identical construction of an automaton $\mathcal{B}_\psi$ that accepts a tree $\mathcal{T}$ if and only if $\mathcal{T}$ is a tree representation (more or less as in Sect 4.1) of a model $\mathfrak{D} \models \psi$.

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, let K be a set of $2k$ constants and let $f : D \rightarrow K$ be an appropriate function naming each element d of $\mathfrak{D}$ by a constant $a_d \in K$ as described in Sect. 4.1. Our automaton works on the description of $\langle T, (\mathfrak{F}(v))_{v \in T} \rangle$ by a labelled tree $\langle T, L \rangle$, where the label of $L(v)$ of a node v is the pair (C, Γ) such that $C \subseteq K$ is the set of constants that occur at the node v, and Γ contains all atomic and negated atomic formulae $\beta(\boldsymbol{a})$ and all guard formulae *clique*$(\boldsymbol{a})$

that are true at node v. It is not difficult to design an alternating automaton that checks whether a given tree is consistently labelled in this sense. One could also integrate the checking of these conditions directly into the automaton $\mathcal{B}_\psi$ constructed below but this would make the construction less transparent.

Let ψ be a sentence in μCGF of vocabulary τ which is in positive normal form (i.e. with negations only in front of atomic formulae). We assume that the fixed point variables in ψ are $T_1, \dots, T_n$, and that for each fixed point variable T_i there is only one formula of the form $[\text{FP } T_i\boldsymbol{x} \,.\, \eta(T_i, \boldsymbol{x})](\boldsymbol{x})$ in ψ (where FP is either LFP or GFP; accordingly T_i is either an LFP-variable or a GFP-variable). Further we assume that for $j > i$, T_j does not preceed T_i in the dependency order of the fixed point variables in ψ, i.e. T_j does not occur free in $[\text{FP } T_i\boldsymbol{x} \,.\, \eta(T_i, \boldsymbol{x})](\boldsymbol{x})$.

Our goal is to describe an automaton $\mathcal{B}_\psi$ that, given a consistently labelled tree $\langle T, L\rangle$ describing a structure $\mathfrak{D}$, checks whether $\mathfrak{D} \models \psi$.

Definition 13. We denote by $cl(\psi)$ the smallest set of formulae such that all subformulae of ψ belong to $cl(\psi)$, and for every relation symbol R occurring in ψ (including equality), $cl(\psi)$ contains formulae $Rx_1 \cdots x_k$ and $\neg Rx_1 \cdots x_k$ with distinct variable symbols $x_1, \dots, x_k$. Further, we put

$$cl(\psi, K) := \{\varphi(\boldsymbol{a}) : \boldsymbol{a} \subseteq K, \varphi(\boldsymbol{x}) \in cl(\psi)\} \cup \{\mathit{true}, \mathit{false}\}.$$

The states of the automaton $\mathcal{B}_\psi$ will be the formulae in $cl(\psi, K)$, with initial state ψ, and the following partition into existential and universal states:

$$\begin{aligned} Q_\exists &= \{\varphi \in cl(\psi, K) : \varphi = \eta \vee \vartheta \text{ or } \varphi = \exists \boldsymbol{x}.\eta\} \cup \{\mathit{false}\} \\ Q_\forall &= cl(\psi, K) - Q_\exists. \end{aligned}$$

The transition function δ is defined as follows.

(1) $\delta(\mathit{true}, (C, \Gamma)) = \delta(\mathit{false}, (C, \Gamma)) = \varnothing$.

(2) If α is a τ-atom or a negated τ-atom then

$$\delta(\alpha, (C, \Gamma)) = \begin{cases} \mathit{true} & \text{if } \alpha \in \Gamma \\ \mathit{false} & \text{if } \alpha \notin \Gamma \end{cases}$$

(3) $\delta(\varphi \vee \vartheta, (C, \Gamma)) = \delta(\varphi \wedge \vartheta, (C, \Gamma)) = \{\varphi, \vartheta\}$.

(4) $\delta([\text{FP } T\boldsymbol{x} \,.\eta(T, \boldsymbol{x})](\boldsymbol{a}), (C, \Gamma)) = \{\eta(T, \boldsymbol{a})\}$.

(5) $\delta(T\boldsymbol{a}, (C, \Gamma)) = \{\eta(T, \boldsymbol{a})\}$ where $[\text{FP } T\boldsymbol{x}.\eta(T, \boldsymbol{x})]$ is the unique fixed point formula in ψ that binds T.

(6) Let φ be of the form $(\exists \boldsymbol{x} \,.\, \mathit{clique}(\boldsymbol{a}, \boldsymbol{x}))\eta(\boldsymbol{a}, \boldsymbol{x})$ or $(\forall \boldsymbol{x} \,.\, \mathit{clique}(\boldsymbol{a}, \boldsymbol{x}))\eta(\boldsymbol{a}, \boldsymbol{x})$. Then

$$\delta(\varphi, (C, \Gamma)) = \begin{cases} \{\eta(\boldsymbol{a}, \boldsymbol{b}) : \boldsymbol{b} \subseteq C, \mathit{clique}(\boldsymbol{a}, \boldsymbol{b}) \in \Gamma\} \cup \{\circ\varphi\} & \text{if } \boldsymbol{a} \subseteq C \\ \varnothing & \text{if } \boldsymbol{a} \not\subseteq C. \end{cases}$$

The acceptance condition Ω is defined by

$$\Omega(\varphi) = \begin{cases} 2i-1 & \text{for } \varphi = T_i\boldsymbol{a} \text{ and } T_i \text{ an LFP-variable} \\ 2i & \text{for } \varphi = T_i\boldsymbol{a} \text{ and } T_i \text{ a GFP-variable} \\ 2n+1 & \text{for } \varphi = \exists \boldsymbol{x} \,.\, \eta \\ 2n+2 & \text{otherwise.} \end{cases}$$

Theorem 11. *Suppose that $\langle T, L\rangle$ represents a tree-decomposition of width $k-1$ of a structure $\mathfrak{D}$. Then $\mathfrak{B}_\psi$ accepts $\langle T, L\rangle$ if and only if $\mathfrak{D} \models \psi$.*

We just try to convey some intuition why this is true. For a detailed proof, we refer to [6].

The game $\mathcal{G}(\mathcal{B}_\psi, \mathcal{T})$ associated with the automaton $\mathfrak{B}_\psi$ on input $\mathcal{T}$ is basically just the *model checking game* for ψ on the structure $\mathfrak{D}$ that is represented by $\mathcal{T}$. (This game generalizes the model checking game for the μ-calculus described for instance in [10]). It is not difficult to see that if player 0 has a strategy to win $\mathcal{G}(\mathcal{B}_\psi, \mathcal{T})$ in a *finite* number of moves, then $\mathfrak{D} \models \psi$, and if player 1 has such a strategy, then $\mathfrak{D} \models \neg\psi$. However, the interesting plays are the infinite ones. We have to distinguish different cases:

(1) Suppose that there are infinitely many occurrences of fixed point formulae $T\boldsymbol{a}$ in the play. Then the 'outermost' fixed point variable T_i (the least with respect to the dependency order) that appears infinitely often determines which player has won. If it is a LFP-variable then the minimal number $\Omega(q)$ appearing infinitely often on the play is odd and therefore the play is winning for player 1. One can show that if player 1 has a strategy to enforce this, then $\mathfrak{D} \models \neg\psi$ since the regeneration sequence of some LFP-formula in ψ is not well-founded. If T_i is a GFP-variable, then player 0 wins and $\mathfrak{D} \models \psi$.

(2) If no fixed point variable appears infinitely often on a infinite play, then there is an existential or universal formula that is never reduced but always 'passed over' to a neighbouring node (see clause (6) of the definition of the automaton). If this formula is existential, say $\exists \boldsymbol{x}.\eta(\boldsymbol{x})$, then by the definition of Ω, player 0 looses. Indeed, if she gets trapped in such a situation then this means that she cannot find a witness $\boldsymbol{a}$ and then win the remaining game from $\eta(\boldsymbol{a})$. Hence $\mathfrak{D} \models \neg\psi$ (recall that ψ is in positive normal form). If the formula is of the form $\forall x.\eta(\boldsymbol{x})$, then player 1 wins. Indeed, such an outcome just means that player 1 does not find a 'bad' $\boldsymbol{a}$ such that he hopes to win on $\eta(\boldsymbol{a})$. In this case $\mathfrak{D} \models \psi$.

Note that the number of states of the automaton B_ψ is bounded by $|\psi|^{2k \log k}$ where k is the width of ψ. Given that the emptiness problem for our automata can be solved in exponential time with respect to the number of states, it follows that the satisfiability problem for guarded fixed point logic is solvable in 2EXPTIME for sentences of unbounded width an in EXPTIME for sentences of bounded width.

References

1. H. Andréka, J. van Benthem, and I. Németi, *Modal languages and bounded fragments of predicate logic*, Journal of Philosophical Logic, 27 (1998), pp. 217–274.
2. J. van Benthem, *Dynamic bits and pieces*, ILLC research report, University of Amsterdam, 1997.
3. A. Emerson and 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.
4. J. Flum, *On the (infinite) model theory of fixed-point logics*, in Models, algebras, and proofs, X. Caicedo and C. Montenegro, eds., no. 2003 in Lecture Notes in Pure and Applied Mathematics Series, Marcel Dekker, 1999, pp. 67–75.
5. E. Grädel, *On the restraining power of guards*, Journal of Symbolic Logic. To appear.
6. E. Grädel and I. Walukiewicz, *Guarded fixed point logic*, in Proc. 14th IEEE Symp. on Logic in Computer Science, 1999.
7. D. Kozen, *Results on the propositional μ-calculus*, Theoretical Computer Science, 27 (1983), pp. 333–354.
8. M. Rabin, *Decidability of second-order theories and automata on infinite trees*, Transactions of the AMS, 141 (1969), pp. 1–35.
9. B. Reed, *Tree width and tangles: A new connectivity measure and some applications*, in Surveys in Combinatorics, R. Bailey, ed., Cambridge University Press, 1997, pp. 87–162.
10. C. Stirling, *Bisimulation, model checking and other games.* Notes for the Mathfit instructional meeting on games and computation. Edinburgh, 1997.
11. M. Vardi, *Why is modal logic so robustly decidable?*, in Descriptive Complexity and Finite Models, N. Immerman and P. Kolaitis, eds., vol. 31 of DIMACS Series in Discrete Mathematics and Theoretical Computer Science, AMS, 1997, pp. 149–184.
12. ———, *Reasoning about the past with two-way automata*, in Automata, Languages and Programming ICALP 98, vol. 1443 of Springer Lecture Notes in Computer Science, 1998, pp. 628–641.
13. W. Zielonka, *Infinite games on finitely coloured graphs with applications to automata on infinite trees*, Theoretical Computer Science, 200 (1998), pp. 135–183.