# Guarded Fixed Point Logic

Erich Grädel[1]
RWTH Aachen

Igor Walukiewicz[2]
Warsaw University

## Abstract

*Guarded fixed point logics are obtained by adding least and greatest fixed points to the guarded fragments of first-order logic that were recently introduced by Andréka, van Benthem and Németi. Guarded fixed point logics can also be viewed as the natural common extensions of the modal $\mu$-calculus and the guarded fragments. We prove that the satisfiability problems for guarded fixed point logics are decidable and complete for deterministic double exponential time. For guarded fixed point sentences of bounded width, the most important case for applications, the satisfiability problem is* EXPTIME*-complete.*

## 1 Introduction

Modal logics are widely used in a number of areas in computer science, in particular for the specification and verification of hardware and software systems, for knowledge representation, in databases, and in artificial intelligence. The most important reason for the successful applications of these logics is that they provide a good balance between expressive power and computational complexity. Indeed, a great number of logical formalisms have successfully been tailored in such a way that they are powerful enough to express interesting properties for a specific application but still admit reasonably efficient algorithms for their central computational problems, in particular for model checking and for satisfiability or validity tests. Many of these formalisms are essentially modal logics although this is not always apparent from their 'official' definitions (as for instance in the case of *description logics* [3]).

The basic propositional (poly)modal logic ML (for a given set $A$ of 'actions' or 'modalities') extends propositional logic by the possibility to construct formulae $\langle a\rangle\psi$ and $[a]\psi$ (where $a \in A$) with the meaning that $\psi$ holds at some, respectively each, $a$-successor of the current state. An equivalent formalism from a different application area is the description logic $\mathcal{ALC}$. Although ML and $\mathcal{ALC}$ are too weak for most of the really interesting applications, they can be extended by features like path quantification, transitive closure operators, counting quantifiers, least and greatest fixed points etc. It has turned out that most of these extensions are still decidable and indeed of considerable practical importance.

Up to now, the reasons for these good algorithmic properties of modal logics have not been sufficiently understood. In [15] Vardi explicitly asked the question: *"Why is modal logic so robustly decidable?"*.

To discuss this question, it is useful to consider propositional modal logic as a fragment of first-order logic. Kripke structures (or equivalently, transition systems), which provide the semantics for modal logics, are relational structures with only unary and binary relations. Every formula $\psi \in$ ML can be translated into a first-order formula $\psi^*(x)$ with one free variable, which is equivalent in the sense 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 $\psi^*(y)$ is obtained from $\psi^*(x)$ by replacing all occurrences of $x$ by $y$ and vice versa and 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. It has turned out that the modal fragment has interesting algorithmic and model-theoretic properties (see [1] and the references given there). On the other side, the modal fragment is a very small part of first-order logic. It is properly contained in $\text{FO}^2$, relational first-order logic with only two variables. But although $\text{FO}^2$ is decidable and has the finite model property (see [10, 6]), it lacks the nice model-theoretic properties [1, 9] and, in particular, the robust decidability properties of modal logics. Indeed while the ex-

[1] Mathematische Grundlagen der Informatik, RWTH Aachen, D-52056 Aachen, graedel@informatik.rwth-aachen.de

[2] Instytut Informatyki UW, Banacha 2, 02-097 Warszawa, igw@mimuw.edu.pl. Supported by Polish KBN grant No. 8 T11C 027 16.

tensions of modal logic by path quantification, transitive closure operators, least and greatest fixed points etc. are still decidable and actually algorithmically quite manageable, most of the corresponding extensions of $\mathrm{FO}^2$ are highly undecidable (see [7, 8]). In particular this is the case for fixed-point logic with two variables, which is the natural common extension of $\mathrm{FO}^2$ and the $\mu$-calculus. The embedding of ML in $\mathrm{FO}^2$ therefore does *not* give a satisfactory answer to Vardi's question.

An alternative explanation for the good properties of modal logics has been proposed by Andréka, van Benthem and Németi [1]. Starting from the observation that in the translation of modal formulae into first-order formulae, the quantifiers are used only in a very restricted way, they defined the *guarded fragment* of first-order logic. They dropped the restriction to use only two variables and only monadic and binary predicates, but imposed that all quantifiers must be relativized by atomic formulae. This means that quantifiers appear only in the form

$$\begin{array}{ll}\exists \boldsymbol{y}(\alpha(\boldsymbol{x},\boldsymbol{y}) \wedge \psi(\boldsymbol{x},\boldsymbol{y})) & \text{or} \\ \forall \boldsymbol{y}(\alpha(\boldsymbol{x},\boldsymbol{y}) \rightarrow \psi(\boldsymbol{x},\boldsymbol{y})). & \end{array}$$

Thus quantifiers may range over a tuple $\boldsymbol{y}$ of variables, but are 'guarded' by an atom $\alpha$ that contains all the free variables of $\psi$.

The guarded fragment GF extends the modal fragment and turns out to have interesting properties [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 $\mathrm{FO}^k$, do hold also for the guarded fragment; (5) The notion of equivalence under guarded formulae can be characterized by a straightforward generalization of bisimulation.

In a further paper, van Benthem [2] generalized the guarded fragment to the *loosely guarded fragment* (LGF) where quantifiers are guarded by conjunctions of atomic formulae of certain forms (details will be given in the next section.) Most of the properties of GF generalize to LGF.

In [5] Grädel showed that the the satisfiability problems for GF and LGF are complete for 2EXPTIME, the class of problems solvable by a deterministic algorithm in time $2^{2^{p(n)}}$, for some polynomial $p(n)$.

If it is indeed the case that, as suggested by Andréka, van Benthem and Németi, the guarded nature of quantification in modal logics is the main responsible also for their good algorithmic properties, then we are naturally lead to the following question:

> If we extend the guarded fragments of first-order logic 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 this paper we answer these questions. We show that the model-theoretic and algorithmic methods that are available for the $\mu$-calculus on one side, and the guarded fragments of first-order logic on the other side, can be combined and generalized to provide positive results for guarded fixed point logic. In fact we can give precise complexity bounds.

**Theorem 1.1.** *The satisfiability problem for guarded fixed point logic is* 2EXPTIME*-complete.*

Note that this is the same complexity as for guarded first-order sentences, so we essentially do not pay any penalty for fixed points! Even so, double exponential time is of course a very high complexity level, which is usually beyond practical possibilities. However, the reason for this doubly exponential complexity is just the fact that the formulae have unbounded *width*, i.e., they may contain subformulae with an unbounded number of free variables. Given that even a single predicate of arity $n$ over a domain of just two objects leads to $2^{2^n}$ possible types already on the atomic level, the double exponential lower complexity bound is hardly a surprise.

Fortunately, in most practical applications, formulae have only bounded width. In particular, for a fixed finite vocabulary all guarded formulae have bounded width. For example, the translation of the $\mu$-calculus into guarded fixed point logic uses at most binary relations and leads to formulae of width two. Our proof shows that guarded fixed point sentences of bounded width have only single exponential complexity.

**Theorem 1.2.** *The satisfiability problem for guarded fixed point sentences of bounded width is* EXPTIME*-complete.*

In particular this generalizes the EXPTIME-completeness of the $\mu$-calculus and the $\mu$-calculus with inverse modalities, as well as a number of EXPTIME-completeness results for description logics. Note that EXPTIME is a complexity level we have to live with even for rather modest extensions of ML or $\mathcal{ALC}$. For most of the popular formalisms in automatic verification (in particular CTL, $\mathrm{CTL}^*$ and the $\mu$-calculus) and for many description logics the common reasoning problems, in particular the satisfiability problems, are EXPTIME-hard. And indeed, we can often cope with this level of complexity since it is in terms of the length of the formulae, which in many practical applications tend be rather small compared to the often huge size of the structures (i.e., the transition systems or knowledge bases) on which the formulae are evaluated.

Hence, the guarded nature of quantification does indeed seem to provide a convincing explanation for the good algorithmic and model theoretic properties of modal logics. Further, let us point out that guarded fixed point logic provides considerably more expressive power than the $\mu$-calculus or other modal logics. In particular it is not restricted to unary and binary predicates, it allows the use of equalities and inequalities, and it avoids the rigid distinction between state properties and modalities. Given that most of the successful methods for dealing with modal logics or description logics seem to extend to guarded fixed point logic, we may hope that this logic will turn out to be useful also in practical applications.

Here is the plan of this paper. In Sect. 2 we explain the guarded fragments of first-order logic GF and LGF and the guarded fixed point logics $\mu$GF and $\mu$LGF. Further, we show that guarded fixed point logics do not have the finite model property. In Sect. 3 we explain some technical notions on fixed point formulae, like binding definitions and signatures, which are adapted from the corresponding notions on the propositional $\mu$-calculus [13, 14]. In Sect. 4 we introduce the notion of a tableau for a guarded fixed point sentence. Essentially, a tableau is a tree representation of a model of a sentence. In Sect. 5 we will introduce a particular variant of alternating two way automata designed to accept tableaux. Then we will reduce the satisfiability problem to the emptiness problem for these automata

## 2 Guarded fixed point logic

**Definition 2.1.** The *guarded fragment* GF of first-order logic is defined inductively as follows:

(1) Every relational atomic formula belongs to GF.

(2) GF is closed under propositional connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$ and $\leftrightarrow$.

(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 free$(\psi) \subseteq$ free$(\alpha) = \boldsymbol{x} \cup \boldsymbol{y}$, then the formulae

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

belong to GF.

Here free$(\psi)$ means the set of free variables of $\psi$. An atom $\alpha(\boldsymbol{x}, \boldsymbol{y})$ that relativizes a quantifier as in rule (3) is the *guard* of the quantifier. Notice that the guard must contain *all* the free variables of the formula in the scope of the quantifier.

While the guarded fragment clearly contains the modal fragment of first-order logic, it seems not to be able to express all of temporal logic over $(\mathbb{N}, <)$. Indeed, the straightforward translation of $(\psi$ **until** $\varphi)$ into first-order logic

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

is not guarded in the sense of Definition 2.1. 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 2.2.** The *loosely guarded fragment* LGF is defined similarly to 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

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

belong to LGF, provided that free$(\psi) \subseteq$ free$(\alpha) = \boldsymbol{x} \cup \boldsymbol{y}$ and for every quantified variable $y \in \boldsymbol{y}$ and every variable $z \in \boldsymbol{x} \cup \boldsymbol{y}$ there is at least one atom $\alpha_j$ that contains both $y$ and $z$.

In the translation of $(\psi$ **until** $\varphi)$ 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.

**Notation.** We will 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 $\alpha$ is indeed a proper guard as specified by condition (3) or (3)'.

**Definition 2.3.** The guarded fixed point logics $\mu$GF and $\mu$LGF are obtained by adding to GF and LGF, respectively, the following rules for constructing fixed-point formulae:

Let $W$ be a $k$-ary relation variable and let $\boldsymbol{x} = x_1, \ldots, x_k$ be a $k$-tuple of distinct variables. Further, let $\psi(W, \boldsymbol{x})$ be a guarded formula where $W$ appears only positively and not in guards. Moreover we require that all the free variables of $\psi(W, \boldsymbol{x})$ are contained in $\boldsymbol{x}$. For such a formula $\psi(W, \boldsymbol{x})$ we can build the formulae

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

The parts in square brackets, i.e. $[\text{LFP } W\boldsymbol{x}\,.\,\psi]$ and $[\text{GFP } W\boldsymbol{x}\,.\,\psi]$, are called *fixed point predicates*.

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

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

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

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

and similarly for the greatest fixed points.

**Notation.** We write $[\text{FP } W\boldsymbol{x}\,.\,\psi]$ when we do not want to specify whether we talk about a least or a greatest fixed point.

**Proviso.** We will always assume that our formulae are in negation normal norm, i.e., that all negations are pushed through to the atoms so that negation signs are only in front of atomic formulae.

Please observe that we do not allow to use fixed point predicates in guards. Otherwise guarded quantification would be as powerful as unrestricted quantification. Indeed, for every $k$, we can define the universally true $k$-ary relation by the fixed point predicate $[\text{GFP } U^k x_1 \cdots x_k\,.\,\mathit{true}]$ (where $\mathit{true}$ stands for any tautology). Using these predicates as guards one could obtain unrestricted quantification. Also the use of the fixed point variable $W$ as a guard inside the formula defining it at as a least or greatest fixed point, or the use of additional first-order variables as parameters in fixed point formulae would lead to an undecidable logic, as has been shown by Martin Otto.

**Infinity axioms.** Contrary to both GF and the $\mu$-calculus, guarded fixed-point logic does not have the finite model property. An infinity axiom is a satisfiable sentence that does not have a finite model.

**Proposition 2.4.** *Guarded least fixed point logic (even with only two variables, without nested fixed points and without equality) contains infinity axioms.*

*Proof.* Consider the formulae

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

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 these formulae is infinite. On the other side, the formulae are clearly satisfiable, for instance by $(\omega, <)$. $\square$

# 3 Signatures of fixed point formulae

A formula is *well-named* if every fixed-point variable is bound at most once in the formula and free second-order variables are distinct from bound variables. Obviously every formula is equivalent to a well-named one. Let us fix in this section a well-named sentence $\psi$ of $\mu$LGF.

**Definition 3.1.** If $T$ is bound in $\psi$ then the *binding definition* of $T$ in $\psi$ is the (unique) fixpoint predicate in $\psi$ of the form $[\text{FP } T\boldsymbol{z}\,.\,\eta(T, \boldsymbol{z})]$. The *definition list* for $\psi$ is the function $\mathcal{D}_\psi$ assigning to each fixed point variable in $\psi$ its binding definition. A variable $T$ is called an LFP-*variable* if $\mathcal{D}_\psi(T)$ is an LFP-predicate; similarly we define GFP-variables. Let $\lessdot_\psi$ be a binary relation on variables bound in $\psi$ defined by $T \lessdot_\psi T'$ iff $T$ occurs free in $\mathcal{D}_\psi(T')$.

**Lemma 3.2.** *The transitive closure of $\lessdot_\psi$ is a partial order.*

*Proof.* It is enough to show that this transitive closure is asymmetric. This follows because whenever $T \lessdot_\psi T'$, then $T \neq T'$ and $\mathcal{D}_\psi(T')$ is properly included in $\mathcal{D}_\psi(T)$. $\square$

**Definition 3.3.** A *dependency order* for a well named formula $\psi$ is a linear order $<_\psi$ that extends $\lessdot_\psi$.

**Example.** It may be the case that $X <_\psi Z$ although $X$ does not occur free in $\mathcal{D}_\psi(Z)$. Consider:

$$\psi = \big[\text{LFP } Xz_1.[\text{GFP } Yz_2.Xz_2 \vee [\text{LFP } Zz_3.Yz_3 \vee Zz_3](z_2)](z_1)\big](a)$$

We have $X \lessdot_\psi Y$ and $Y \lessdot_\psi Z$ hence $X <_\psi Z$.

**Example.** It is not the case that $X <_\psi Y$ whenever $\mathcal{D}_\psi(Y)$ is a subformula of $\mathcal{D}_\psi(X)$. For the formula:

$$\psi = \big[\text{LFP } Xz_1.\ Xz_1 \vee [\text{LFP } Yz_2.\ Yz_2](z_1)\big](a)$$

we have that $X$ and $Y$ are incomparable in $\lessdot_\psi$ as $X$ does not occur free in $\mathcal{D}_\psi(Y) = [\text{LFP } Yz_2.Yz_2]$. Because $<_\psi$ is an arbitrary extension of $\lessdot_\psi$ it may be the case that $Y <_\psi X$.

Given a formula $\eta$ with a free $k$-ary second-order variable $T$ and a formula $\gamma(\boldsymbol{x})$ with free variables $\boldsymbol{x} = x_1, \ldots, x_k$, we write $\eta[\gamma/T]$ for the formula obtained by replacing all occurrences of atoms $T\boldsymbol{y}$ in $\eta$ by $\gamma(\boldsymbol{y})$.

**Definition 3.4.** For all ordinals $\alpha$, the *approximations* $\eta^\alpha(\boldsymbol{x})$ of an LFP-predicate $[\text{LFP } T\boldsymbol{x}\,.\,\eta(T, \boldsymbol{x})]$ are defined inductively in the usual way. Let $\eta^0(\boldsymbol{x}) := \mathit{false}$; if $\alpha$ is a limit ordinal, let $\eta^\alpha(\boldsymbol{x}) := \bigvee_{\beta<\alpha} \eta^\beta(\boldsymbol{x})$; finally let

$\eta^{\alpha+1}(\boldsymbol{x}) = \eta([\eta^\alpha/T], \boldsymbol{x})$ for every $\alpha$. (Note that $\eta^\alpha(\boldsymbol{x})$ is in general *not* a formula in $\mu$LGF since the ordinals need not be finite. However, the approximations of the fixed point predicates are contained in a guarded infinitary logic that extends $\mu$LGF by conjunctions and disjunctions over arbitrary sets of formulae.)

Let $\mathcal{D}_\psi$ be a definition list for $\psi$ and let $T_1 <_\psi T_2 <_\psi \cdots <_\psi T_n$ be the enumeration of the fixed point variables of $\psi$ with respect to an arbitrary, but fixed dependency order for $\psi$. Note that all free fixed-point variables in $\mathcal{D}_\psi(T_i)$ are among $T_1, \dots, T_{i-1}$ and (due to the restrictions on the use of fixed-point operators in Definition 2.3) the predicate $\mathcal{D}_\psi(T_i)$ has no free first-order variables.

**Definition 3.5.** Let $\varphi$ be a formula in $\mu$LGF without free first order variables and whose free second order variables belong either to the vocabulary of $\psi$ or to the domain of $\mathcal{D}_\psi$. For an $n$-tuple of ordinals $\vec{\tau} = (\tau_1, \dots, \tau_n)$ we define the sentence $\varphi^{\vec{\tau}}$ by:

$$\varphi[\mathcal{D}^\dagger(T_n)/T_n][\mathcal{D}^\dagger(T_{n-1})/T_{n-1}]\cdots[\mathcal{D}^\dagger(T_1)/T_1]$$

where $\mathcal{D}^\dagger(T) = \mathcal{D}(T)$ if $T$ is a GFP variable and $\mathcal{D}^\dagger(T_i) = \eta^{\tau_i}$ if $T_i$ is an LFP-variable with $\mathcal{D}(T_i) = [\text{LFP } T_i\boldsymbol{x} \,.\, \eta(T, \boldsymbol{x})]$.

The LFP-*s*ignature of $\varphi$ in $\mathfrak{A}$, denoted $\text{Sig}_\mu(\varphi, \mathfrak{A})$, is the smallest tuple of ordinals $\vec{\tau}$ (with respect to the lexicographical ordering) such that $\mathfrak{A} \models \varphi^{\vec{\tau}}$.

Please observe that a signature $\text{Sig}_\mu(\varphi, \mathfrak{A})$ is defined if and only if $\mathfrak{A} \models \varphi[\mathcal{D}(T_n)/T_n]\cdots[\mathcal{D}(T_1)/T_1]$. Further the definition implies that $\text{Sig}_\mu(\varphi, \mathfrak{A})$ is 0 on positions corresponding to GFP-variables.

The following technical lemma about signatures will be useful in the proofs of correctness of our constructions.

**Lemma 3.6.** *The following facts hold for signatures (assuming that in all formulas under consideration all free fixed- point variables are in the domain of $\mathcal{D}_\psi$):*

*(i)* *If $\varphi \vee \eta$ has signature $\vec{\tau}$ in $\mathfrak{A}$ then either $\varphi$ or $\eta$ has signature $\vec{\tau}$ in $\mathfrak{A}$.*

*(ii)* *If $\varphi \wedge \eta$ has signature $\vec{\tau}$ in $\mathfrak{A}$ then both $\varphi$ and $\eta$ have signature not bigger than $\vec{\tau}$ in $\mathfrak{A}$.*

*(iii)* *If $(\forall \boldsymbol{x} \,.\, \alpha(\boldsymbol{x}))\varphi(\boldsymbol{x})$ has signature $\vec{\tau}$ in $\mathfrak{A}$ then for every tuple $\boldsymbol{a}$ such that $\mathfrak{A} \models \alpha(\boldsymbol{a})$, the sentence $\varphi(\boldsymbol{a})$ has signature not bigger than $\vec{\tau}$.*

*(iv)* *If $(\exists \boldsymbol{x} \,.\, \alpha(\boldsymbol{x}))\varphi(\boldsymbol{x})$ has signature $\vec{\tau}$ in $\mathfrak{A}$ then there is a tuple $\boldsymbol{a}$ such that $\mathfrak{A} \models \alpha(\boldsymbol{a})$ and $\varphi(\boldsymbol{a})$ has signature $\vec{\tau}$ in $\mathfrak{A}$.*

*(v)* *If $[\text{FP } T_i\boldsymbol{x} \,.\, \eta(T_i, \boldsymbol{x})](\boldsymbol{a})$ has signature $\vec{\tau}$ then $\eta(T_i, \boldsymbol{a})$ has the same signature on the positions $1, \dots, i-1$.*

*(vi)* *If $T_i\boldsymbol{a}$ has signature $\vec{\tau}$ and $\mathcal{D}(T_i) = [\text{LFP } T_i\boldsymbol{x} \,.\, \eta(T_i, \boldsymbol{x})]$ then $\eta(T_i, \boldsymbol{a})$ has a strictly smaller signature and the difference is on the first $i$ positions.*

*(vii)* *If $T\boldsymbol{a}$ has signature $\vec{\tau}$ and $\mathcal{D}(T) = [\text{GFP } T\boldsymbol{x} \,.\, \eta(T, \boldsymbol{x})]$ then $\eta(T, \boldsymbol{a})$ has also signature $\vec{\tau}$.*

*Proof.* We will consider only the case of an LFP-variable. Let $T_i$ be the $i$-th variable in the $<_\psi$ ordering and let $D(T_i) = [\text{LFP } T_i\boldsymbol{x} \,.\, \eta(T_i, \boldsymbol{x})]$. Suppose that $(\tau_1, \dots, \tau_n)$ is the signature of $T_i\boldsymbol{a}$ in $\mathfrak{A}$. Observe that that $T_k$ does not appear in $\mathcal{D}(T_j)$ if $j < k$. Hence by the definition of the signature we get:

$$\mathfrak{A} \models \eta^{\tau_i}[\mathcal{D}^\dagger(T_{i-1})/T_{i-1}]\cdots[\mathcal{D}^\dagger(T_1)/T_1](\boldsymbol{a}) \qquad (1)$$

where $\mathcal{D}^\dagger(T_j)$ is as in Definition 3.5. We know that $\tau_i$ is the least ordinal such that (1) is satisfied. So in particular it is a successor ordinal. By the definition of the approximation we have

$$\mathfrak{A} \models \eta[\eta^{\tau_i - 1}/T_i][\mathcal{D}^\dagger(T_{i-1})/T_{i-1}]\cdots[\mathcal{D}^\dagger(T_1)/T_1](\boldsymbol{a}).$$

Hence the signature of $\eta(T_i, \boldsymbol{a})$ is not bigger than $(\tau_1, \dots, \tau_{i-1}, \tau_i - 1)$ on the first $i$ positions. $\square$

## 4 Tableaux for $\mu$LGF

A tableau for $\psi$ will be a tree labelled with $\psi$-types which we are going to define now.

First we define a closure $cl(\psi)$ of the formula $\psi$ to be the smallest set of formulae such that

*(i)* All subformulas of $\psi$ belong to $cl(\psi)$.

*(ii)* for every relation symbol $R$ occurring in $\psi$, $cl(\psi)$ contains formulae $Rx_1 \cdots x_k$ and $\neg Rx_1 \cdots x_k$ with distinct variable symbols $x_1, \dots, x_k$.

**Definition 4.1.** For any set $C$ of constants, let:

$$\begin{aligned} cl(\psi, C) := \{\varphi(\boldsymbol{a}) : \boldsymbol{a} \subseteq C, \varphi(\boldsymbol{x}) \in cl(\psi)\} \cup \\ \{a = a, \neg(a = b) : a, b \in C, a \neq b\} \end{aligned}$$

A $(\psi, C)$*-type* is a subset $\Gamma$ of $cl(\psi, C)$ with the following properties:

(a) $\{a = a : a \in C\} \cup \{\neg(a = b) : a, b \in C, a \neq b\} \subseteq \Gamma$.

(b) For all atomic $\alpha \in cl(\psi, C)$ either $\alpha \in \Gamma$ or $\neg\alpha \in \Gamma$, but not both.

(c) If $\varphi \vee \vartheta \in \Gamma$ then $\varphi \in \Gamma$ or $\vartheta \in \Gamma$.

(d) If $\varphi \wedge \vartheta \in \Gamma$ then $\varphi \in \Gamma$ and $\vartheta \in \Gamma$.

(e) If $(\forall \boldsymbol{x} \,.\, \alpha(\boldsymbol{a}, \boldsymbol{x}))\varphi(\boldsymbol{a}, \boldsymbol{x}) \in \Gamma$ and for some $\boldsymbol{b} \subseteq C$ we have $\alpha(\boldsymbol{a}, \boldsymbol{b}) \in \Gamma$, then also $\varphi(\boldsymbol{a}, \boldsymbol{b}) \in \Gamma$.

(f) If $[\mathrm{FP}\ T\boldsymbol{x} \,.\, \eta(T, \boldsymbol{x})](\boldsymbol{a}) \in \Gamma$ then also $\eta(T, \boldsymbol{a}) \in \Gamma$.

(g) If $T\boldsymbol{a} \in \Gamma$ and $\mathcal{D}_\psi(T) = [\mathrm{FP}\ T\boldsymbol{x} \,.\, \eta(T, \boldsymbol{x})]$ then $\eta(T, \boldsymbol{a}) \in \Gamma$.

Let $\mathit{Types}(\psi, C)$ denote the set of all the $(\psi, C)$-types.

**Definition 4.2.** A *tableau* over a set of constants $K$ for a sentence $\psi \in \mu$-LGF is a tree $\mathcal{T}$ labelled with pairs $(\Gamma, C)$ where $C \subseteq K$ and $\Gamma$ is a $(\psi, C)$-type. $\mathcal{T}$ is constructed inductively according to the following rules:

(1) The root is labelled by $(\Gamma, \varnothing)$ where $\Gamma$ is a $(\psi, \varnothing)$-type containing $\psi$.

(2) Suppose that a node $v$ of $\mathcal{T}$ is labelled by $(\Gamma, C)$ and that there is an existential sentence $\exists \boldsymbol{x} \,.\, \varphi(\boldsymbol{a}, \boldsymbol{x})$ in $\Gamma$ such that for all $\boldsymbol{b} \subseteq \boldsymbol{a}$ we have $\neg\varphi(\boldsymbol{a}, \boldsymbol{b}) \in \Gamma$. In this case we choose a tuple $\boldsymbol{b} \subseteq \boldsymbol{a} \cup (K \setminus C)$ and construct a child $w$ of $v$ labelled by $(\Gamma', \boldsymbol{a} \cup \boldsymbol{b})$ where $\Gamma'$ is a $(\psi, \boldsymbol{a} \cup \boldsymbol{b})$-type that contains the sentence $\varphi(\boldsymbol{a}, \boldsymbol{b})$ and that also contains all the sentences of $\Gamma$ referring only to constants from $\boldsymbol{a}$.

## 5 Two-way alternating automata on trees (of arbitrary degree)

Automata play a very important role for the satisfiability testing and model checking of modal logics. In particular, *alternating tree automata* seem to be the right model to get optimal complexity bounds for the $\mu$-calculus and its relatives. In [16] Vardi used alternating *two-way* tree automata (A2A) to establish decidability and EXPTIME-completeness of the $\mu$-calculus with backward modalities. His model of A2A works on trees of bounded branching where nodes have outgoing arcs labelled by indices $1, 2, \ldots, k$. The values of the transition function are positive Boolean formulae over pairs $(q, i) \in Q \times \{-1, 0, 1, \ldots, k\}$. A pair $(q, i)$ means that the automaton assumes state $q$ and proceeds into direction $i$. Here the direction 0 means that the automaton stays at the current node and direction -1 means that the automaton goes to the parent node. Vardi proved that the emptiness problem for his A2A can be solved in exponential time with respect to the number of states of the automaton.

To obtain a decision procedure for the $\mu$-calculus with backward modalities Vardi first proves, adapting techniques of Streett and Emerson [14], that this logic has the *tree model property*: every satisfiable sentence has a tree model with bounded degree. Then he shows, that for every sentence $\psi$ one can build an alternating automaton $\mathcal{A}_\psi$ that accepts a tree of bounded degree iff it is a model for $\psi$.

We will use here a different model of alternating automata that work on trees of arbitrary, finite or infinite, degree. While the trees will have directed edges (from fathers to sons), the automata will not distinguish this orientation and will be able to proceed to any neighbour of a node, i.e., either to the father of the current node or to any son. A general forgetful determinacy theorem for graph games can be used to reduce the emptiness problem for our automata to the emptiness problem for Vardi's automata.

**Definition 5.1.** 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, $\Sigma$ is an input alphabet and $q_0 \in Q$ is an initial state. The transition function has the form

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

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 $\Omega : Q \to \mathbb{N}$ specifies the acceptance condition of the automaton.

We will define a run of our automata in terms of parity games. For this we will first introduce general concepts concerning such games.

A *parity game* is a tuple $\mathcal{G} = \langle V_0, V_1, E, \Omega \rangle$ where $V = V_0 \cup V_1$ is a, possibly infinite, set of positions; $E \subseteq V \times V$ is an edge relation and $\Omega : V \to \mathbb{N}$ is a function defining *parity winning condition*. A game is played between two players (player 0 and 1). A move of the game consists of moving a token from one position to the other along an edge of the game graph. If a token is in a vertex from $V_0$ then player 0 makes the move, otherwise player 1 moves the token. The result of a play from an initial position $v_0$ is a finite or an infinite sequence of positions $v_0, v_1, \ldots$ This sequence can be finite only because one of the players cannot make a move. In this case he looses. If the play is infinite we look at the sequence of numbers $\Omega(v_0), \Omega(v_1), \ldots$ Player 0 wins iff this sequence satisfies the *parity condition* given by $\Omega$, i.e., the smallest number among the numbers appearing infinitely often in the sequence is even.

A *strategy* for player 0 in $\mathcal{G}$ is a partial function $f : V^* \times V_0 \to V$ such that whenever $f(\vec{v}v)$ is defined then $(v, f(\vec{v}v))$ is an edge from $E$. A finite path $v_0 v_1 \cdots v_n$ is *consistent* with $f$ if for every $i = 0, \ldots, n-1$ with $v_i \in V_0$ we have $v_{i+1} = f(v_0 \cdots v_i)$. An infinite path is consistent with $f$ if every its finite prefix is. A strategy $f$ is *winning* from a position $v_0$ if:

1. every consistent path starting in $v_0$ and ending in a position from $V_0$ can be prolonged to a consistent path, and

2. every infinite consistent path starting in $v_0$ is winning for player 0.

We can now go back to our tree automata. Let $\mathcal{T} = \langle T, L : T \to \Sigma\rangle$ be a tree labelled by symbols from $\Sigma$. We will use games to define the notion of acceptance. For a given automaton $\mathcal{A}$ and a tree $\mathcal{T}$ let $\mathcal{G}(\mathcal{A},\mathcal{T}) = \langle V_0, V_1, E, \Omega\rangle$ be parity game 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 $v$.
- $\Omega(v,q) = \Omega(q)$.

We say that $\mathcal{A}$ *accepts* $\mathcal{T}$ iff player 0 has a winning strategy from the position $(\lambda, q_0)$ where $\lambda$ is the root of $\mathcal{T}$ and $q_0$ is the initial state of $\mathcal{A}$.

Our first goal is to show that if an automaton accepts some tree then it accepts a tree of a degree bounded by some function in the number of states of the automaton. For this we will use a *memoryless determinacy theorem* for parity games [4, 11, 17].

A strategy $f$ in a parity game $\mathcal{G}$ is called memoryless if $f(\vec{v}_1 v_1) = f(\vec{v}_2 v_2)$ whenever $v_1 = v_2$; in other words a memoryless strategy depends only on the current position of the play and not on its history.

**Theorem 5.2 (Emerson, Jutla).** *For every parity game there exist a memoryless winning strategy for player* 0 *that is defined for every position from which there is a winning strategy for player* 0.

As a consequence we can derive the following result.

**Theorem 5.3.** *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.*

*Proof.* Suppose that $\mathcal{A}$ accepts $\mathcal{T}$. Then player 0 has a memoryless winning strategy in $\mathcal{G}(\mathcal{A},\mathcal{T})$ from position $(\lambda, q_0)$. This strategy can be presented as a labelling $f : T \to (Q_\exists \to T \times Q)$ assigning to each node $v$ of $T$ the function $f_v : q \mapsto f(v,q)$. For each node $v$ of $T$ consider the set of relevant vertices for $v$:

$$\begin{aligned} \mathit{Rel}(v) = \{w : \exists q \in Q_\exists \text{ such that} \\ f(v,q) = (w,q') \text{ for some } q' \in Q\} \end{aligned}$$

Clearly $|\mathit{Rel}(v)| \le |Q|$ for all nodes $v$. Now consider the restriction $\mathcal{T}'$ of $\mathcal{T}$ obtained by starting at the root and keeping only the relevant children of each node. One can easily see that $f$ restricted to $\mathcal{T}'$ defines a memoryless winning strategy for player 0 in the game $\mathcal{G}(\mathcal{A},\mathcal{T}')$. Hence $\mathcal{T}'$ is accepted by $\mathcal{A}$. By definition vertices of $\mathcal{T}'$ have degrees bounded by $|Q|$. $\square$

For input trees of bounded degree our automata can easily be translated into alternating two-way automata according to Vardi's definition and the translation at most doubles the number of states. Since the emptiness problem for Vardi's automata is in EXPTIME, we get via Theorem 5.3 the same result for our automata.

**Theorem 5.4.** *The emptiness problem for alternating two-way automata (on trees with arbitrary branching) can be decided in* EXPTIME.

# 6 The satisfiability test

In this section we will construct an automaton $\mathcal{A}_\psi$ accepting exactly those tableaux for $\psi$ that represent a model for $\psi$. Hence $\psi$ will be satisfiable iff there is tableau accepted by $\mathcal{A}_\psi$.

**Definition 6.1.** Let $\psi$ be a well named $\mu$LGF sentence of vocabulary $\tau$, let $\mathcal{D}_\psi$ be its definition list and $T_1 <_\psi \cdots <_\psi T_n$ a dependency ordering on the fixed point variables of $\psi$. For a set of constants $K$ we define an automaton

$$\mathcal{A}_\psi = \langle Q_\exists, Q_\forall, \bigcup_{C \subseteq K} (\mathit{Types}(\psi, C) \times \{C\}), \psi, \delta, \Omega\rangle$$

where

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

and $\delta$ is defined as follows.

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

(2) If $\alpha$ is a $\tau$-atom or a negated $\tau$-atom then

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

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

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

(5) $\delta(T\boldsymbol{a}, C) = \{\eta(T,\boldsymbol{a}), C)\}$ where $\mathcal{D}(T) = [\text{FP } T(\boldsymbol{x}).\eta(T,\boldsymbol{x})]$.

(6) Let $\varphi$ be of the form $(\exists \boldsymbol{x} \,.\, \alpha(\boldsymbol{a}, \boldsymbol{x}))\eta(\boldsymbol{a}, \boldsymbol{x})$ or $(\forall \boldsymbol{x} \,.\, \alpha(\boldsymbol{a}, \boldsymbol{x}))\eta(\boldsymbol{a}, \boldsymbol{x})$. If $\boldsymbol{a} \subseteq C$, then

$$\delta(\varphi,(\Gamma,C)) = \{\eta(\boldsymbol{a},\boldsymbol{b}) : \boldsymbol{b} \subseteq C, \alpha(\boldsymbol{a},\boldsymbol{b}) \in \Gamma\} \cup \{\bigcirc\varphi\}$$

If $\boldsymbol{a} \not\subseteq C$, then $\delta(\varphi,(\Gamma,C)) = \varnothing$.

Finally $\Omega$ is defined by

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

The size of the automaton $\mathcal{A}_\psi$ is linear in the size of $\mathit{Types}(\psi,K)$. In turn, the size of this set depends linearly on the size of $\psi$ and exponentially on the size of $K$. As we will see the required number of constants in $K$ will depend linearly on a parameter we call the width of a formula:

**Definition 6.2.** The *width* of $\psi$ is

$$\mathrm{width}(\psi) := \max\{|\mathrm{free}(\varphi)| : \varphi \in cl(\psi)\}.$$

Note that for each sentence $\psi$ from GF or $\mu$GF the width is bounded by the maximal arity of the relation symbols in $\psi$. However, for loosely guarded sentences the width may be larger than the arity of the relation symbols.

**Lemma 6.3.** *Let $\mathfrak{D}$ be a structure and $\psi$ a well-named $\mu$LGF sentence of width $r$. Let $K$ be a set of at least $2r$ constants. If $\mathfrak{D} \models \psi$ then there is a tableau $\mathcal{T}$ for $\psi$ over $K$ that is accepted by $\mathcal{A}_\psi$.*

*Proof.* Let $\mathfrak{D}$ be a model for $\psi$ with the universe $D$. We will construct a tableau for $\psi$ accepted by $\mathcal{A}_\psi$. Given a sentence $\varphi$, possibly with some constants $\boldsymbol{a} \subseteq K$, and a valuation $\chi : \boldsymbol{a} \to D$ we denote by $\mathrm{Sig}^\chi(\varphi)$ the LFP-signature of $\varphi$ in $\mathfrak{D}$ extended with the interpretation of constants from $\boldsymbol{a}$ as given by $\chi$. We construct a tableau

$$\mathcal{T} = \langle T, L : T \to \bigcup_{C\subseteq K} (\mathit{Types}(\psi,C) \times \{C\})\rangle$$

and a function $\chi$ assigning to each node $w \in T$ a valuation of constants from the label of $w$.

Label the root $\lambda$ of the tableau with $(\Gamma,\varnothing)$, where $\Gamma$ is the $(\psi,\varnothing)$-type realized in $\mathfrak{D}$. The function $\chi(\lambda)$ gives the empty valuation as there are no constants.

Suppose now that we have a node $w$ labelled by $(\Gamma,C)$ and a function $\chi = \chi(w) : C \to D$ giving the meaning to each constant in $C$. Suppose moreover that $\Gamma$ is such that $\mathfrak{D},\chi \models \Gamma$. Take an existential sentence $\beta(\boldsymbol{a}) := \exists\boldsymbol{x}\,.\eta(\boldsymbol{a},\boldsymbol{x}) \in \Gamma$ such that for every tuple $\boldsymbol{b} \subseteq \boldsymbol{a}$ we have either $\mathfrak{D},\chi \models \neg\eta(\boldsymbol{a},\boldsymbol{b})$ or $\mathrm{Sig}^\chi(\beta(\boldsymbol{a})) < \mathrm{Sig}^\chi(\eta(\boldsymbol{a},\boldsymbol{b}))$. Since $\mathfrak{D},\chi \models \beta(\boldsymbol{a})$, there is an extension of $\chi$ to $\chi' : (\boldsymbol{a} \cup \boldsymbol{x}) \to D$ such that $\mathfrak{D},\chi' \models \eta(\boldsymbol{a},\boldsymbol{x})$ and moreover $\mathrm{Sig}^\chi(\beta(\boldsymbol{a})) = \mathrm{Sig}^{\chi'}(\eta(\boldsymbol{a},\boldsymbol{x}))$. Let $\{d_1,\dots,d_k\} = \chi'(\boldsymbol{a}\cup\boldsymbol{x}) \setminus \chi(\boldsymbol{a})$ be all the elements of the model used in $\chi'$ that are different than the meanings of the constants in $\boldsymbol{a}$. By our assumption this set is nonempty. We take constants $\boldsymbol{b} = (b_1,\dots,b_k)$ not appearing in the label of $w$, i.e., $\boldsymbol{b} \subseteq K \setminus C$. This is possible by our assumption on the size of $K$. We define the function $\chi'' : (\boldsymbol{a}\cup\boldsymbol{b}) \to D$ by

$$\chi''(b_i) := d_i \quad \text{and} \quad \chi''(a) := \chi(a) \text{ for } a \in \boldsymbol{a}.$$

Let $\Gamma'$ be the unique $(\psi,\boldsymbol{a}\cup\boldsymbol{b})$-type such that $\mathfrak{D},\chi'' \models \Gamma'$. We create a son $w'$ of $w$ labelled with $(\Gamma',\boldsymbol{a}\cup\boldsymbol{b})$ and put $\chi(w') = \chi''$.

To check that the constructed tableau $\mathcal{T}$ is accepted by $\mathcal{A}_\psi$ we define a strategy $f$ for player 0 from $(\lambda,\psi)$ in the game $\mathcal{G}(\mathcal{A}_\psi,\mathcal{T})$.

Given a node $(w,\varphi\vee\vartheta)$ such that $\mathfrak{D},\chi(w) \models \varphi\vee\vartheta$ we define $f(w,\varphi\vee\vartheta) = (w,\varphi)$ if $\mathrm{Sig}^{\chi(w)}(\varphi) \le \mathrm{Sig}^{\chi(w)}(\vartheta)$ and $f(w,\varphi\vee\vartheta) = (w,\vartheta)$ otherwise.

For $\vartheta = \exists\boldsymbol{x}\,.\,\eta(\boldsymbol{a},\boldsymbol{x})$ consider a node $(w,\vartheta)$ with $L(w) = (\Gamma,C)$ and $\mathfrak{D},\chi(w) \models \vartheta$. We define $f(w,\vartheta) = (w,\eta(\boldsymbol{a},\boldsymbol{b}))$ if there is $\boldsymbol{b} \subseteq C$ such that $\mathrm{Sig}^{\chi(w)}(\eta(\boldsymbol{a},\boldsymbol{b})) \le \mathrm{Sig}^{\chi(w)}(\vartheta)$. If it is not the case then we put $f(w,\vartheta) = (w',\vartheta)$, where $w'$ is a son of $w$ such that $\mathrm{Sig}^{\chi(w')}(\eta(\boldsymbol{a},\boldsymbol{b})) \le \mathrm{Sig}^{\chi(w)}(\vartheta)$ (such a $w'$ exists by the construction of $\mathcal{T}$).

To check that the strategy defined above is winning for player 0 let us take any play $(\lambda,\psi),(w_1,\gamma_1),(w_2,\gamma_2),\dots$ played according to this strategy. If the play is finite then it is winning for player 0 because $\gamma_i$ appears in the label of $w_i$ for all $i$.

If the play is infinite then we have two possibilities. It may be the case that after some point, a universal or an existential formula is never reduced, i.e., $\gamma_i = Q\boldsymbol{x}\,.\,\beta$, for all sufficiently large $i$. By the definition of our strategy, if $Q$ is an existential quantifier then $Q\boldsymbol{x}\,.\,\beta$ is reduced in at most two steps. Hence $Q$ must be the universal quantifier and such a play is winning for player 0.

If the play is infinite and every quantifier is eventually reduced then there are infinitely many positions $i$ for which $\gamma_i$ is of the form $T_j\boldsymbol{a}$ for some fixed point variable $T_j$. In this case we say that $T_j$ is *regenerated* at position $i$ in the play. Let $h$ be the smallest among all the indices $j$ such that $T_j$ is regenerated infinitely often on the play. Towards a contradiction suppose that $T_h$ is an LFP-variable. Let us look at the sequence of signatures $\{\mathrm{Sig}^{\chi(w_i)}(\gamma_i,\mathfrak{A})\}_{i=1,2,\dots}$. By Lemma 3.6 and the definition of out strategy, from some point these signatures never increase on positions $\le h$. Moreover the signatures decrease on one of these positions each time $T_h$ is met on the play. This is impossible as a lexicographical ordering of $h$-tuples of ordinals is a well-order. Hence $T_h$ is a GFP-variable and the play is winning for player 0. $\square$

Next we want to show how to construct a model for $\psi$ from a tableau $\mathcal{T}$ for $\psi$ accepted by $\mathcal{A}_\psi$. For a tableau

$$\mathcal{T} = \langle T, L : T \to \bigcup_{C \subseteq K} (\mathit{Types}(\psi, C) \times \{C\})\rangle$$

we define the structure $\mathfrak{A}(\mathcal{T})$. First, for every constant $c \in K$ we call two nodes $v$ and $w$ $c$-equivalent if each node on the path from $v$ to $w$ (including $v$ and $w$) has $c$ in its label. This is an equivalence relation on the set of those nodes in $T$ whose label contains $c$. The carrier of $\mathfrak{A}(\mathcal{T})$ is the set of all $c$-equivalence classes for all $c \in K$. Given equivalence classes $V_1, \ldots, V_n$ for $c_1, \ldots, c_n$-equivalences, respectively, we put $R^{\mathfrak{A}(\mathcal{T})}(V_1, \ldots, V_n)$ iff there is a vertex $v \in \bigcap_{i=1,\ldots,n} V_i$ such that $Rc_1 \cdots c_n$ appears in $L(v)$. From the definition of a tableau we immediately get:

**Lemma 6.4.** *Let $V_1, \ldots V_n$ be a tuple of equivalence classes of constants $c_1, \ldots, c_n$ respectively. If $R^{\mathfrak{A}(\mathcal{T})}(V_1, \ldots, V_n)$ then for every vertex $v \in \bigcap_{i=1,\ldots,n} V_i$ we have $Rc_1 \cdots c_n$ in the label of $v$.*

The following slightly more involved lemma is needed for the case of loosely guarded formulas.

**Lemma 6.5.** *Let $\alpha = \alpha_1(\boldsymbol{x}, \boldsymbol{y}) \wedge \cdots \wedge \alpha_n(\boldsymbol{x}, \boldsymbol{y})$ be a conjunction of atomic formulas such that every variable from $\boldsymbol{y}$ coexists with every variable from $\boldsymbol{x} \cup \boldsymbol{y}$ in some conjunct. If there is a valuation $\chi : \boldsymbol{x} \cup \boldsymbol{y} \to \mathfrak{A}(\mathcal{T})$ such that $\mathfrak{A}(\mathcal{T}), \chi \models \alpha$ and $\bigcap\{\chi(x) : x \in \boldsymbol{x}\} \neq \varnothing$ (as a set of vertices of $\mathcal{T}$) then $\bigcap\{\chi(x) : x \in \boldsymbol{x}\} \cap \bigcap\{\chi(y) : y \in \boldsymbol{y}\} \neq \varnothing$.*

*Proof.* By definition each element of $\mathfrak{A}(\mathcal{T})$ is a subtree of $\mathcal{T}$, hence each $\chi(z)$ is a subtree of $\mathfrak{A}(\mathcal{T})$ for $z \in \boldsymbol{x} \cup \boldsymbol{y}$. We are going to show that for every $z_1, z_2 \in \boldsymbol{x} \cup \boldsymbol{y}$ the intersection $\chi(z_1) \cap \chi(z_2)$ is nonempty. The lemma will then follow from a well-known result in graph theory saying that any collection of pairwise overlapping subtrees of a tree has a common node (see e.g. [12, p. 94]).

If $z_1, z_2 \in \boldsymbol{x}$ then $\chi(z_1) \cap \chi(z_2)$ by assumption of the lemma. Suppose $z_1 \in \boldsymbol{y}$. By the other assumption of the lemma $z_1$ and $z_2$ coexist in some atom of $\alpha$. By definition of $\mathfrak{A}(\mathcal{T})$ this implies that $\chi(z_1) \cap \chi(z_2) \neq \varnothing$. ☐

**Lemma 6.6.** *Let $\mathcal{T}$ be a tableau for $\psi$. If $\mathcal{T}$ is accepted by $\mathcal{A}_\psi$ (over some set of constants $K$) then $\mathfrak{A}(\mathcal{T}) \models \psi$.*

*Proof.* Suppose $\mathfrak{A}(\mathcal{T}) \not\models \varphi$ for some sentence $\varphi$. For $\neg\varphi$ (or rather, an equivalent sentence obtained by pushing the negations to the leaves), the LFP-signature is defined. We can call this signature the GFP-signature of $\varphi$ in $\mathfrak{A}(\mathcal{T})$ and denote it by $\mathrm{Sig}_\nu(\varphi)$. The name comes from the fact that when pushing negation downwards every LFP-operator is changed to GFP and vice versa. For GFP-signatures the dual of Lemma 3.6 holds.

Suppose that $\psi$ is *not* true in $\mathfrak{A}(\mathcal{T})$, hence $\mathrm{Sig}_\nu(\psi)$ is defined. By the assumption that $\mathcal{T}$ is accepted by $\mathcal{A}_\psi$, there is a winning strategy $f$ for player 0 from the position $(\lambda, \psi)$ in $\mathcal{G}(\mathcal{A}_\psi, \mathcal{T})$. Towards a contradiction we will construct a play consistent with $f$ that is winning for player 1.

We start the play from the node $(\lambda, \psi)$. A node $w$ of the tableau labelled by a pair $(\Gamma, C)$ defines a valuation $\chi(w)$ of the constants from $C$ by setting $\chi(w)(c)$ to be the $c$-equivalence class containing $w$. Clearly $\chi(\lambda)$ is the empty function and we have $\mathfrak{A}(\mathcal{T}), \chi(\lambda) \not\models \psi$ from our assumption. Let us denote by $\mathrm{Sig}_\nu^{\chi(w)}(\varphi)$ the GFP-signature of the formula $\varphi$ in the structure $\mathfrak{A}(\mathcal{T})$ extended with constants from $K$ with the meaning given by $\chi(w)$.

Suppose we have constructed our play up to a node labelled by $(w, \vartheta)$ with $L(w) = (\Gamma, C)$ and $\vartheta \in \Gamma$. The way in which we prolong the play depends on the form of $\vartheta$.

*(i)* By the definition of $\mathfrak{A}(\mathcal{T})$ it is impossible that $\vartheta$ is an atomic formula. From Lemma 6.4 it follows that $\vartheta$ cannot be a negation of an atomic formula.

*(ii)* If $\vartheta = \varphi \vee \eta$ then we have $f(w, \vartheta) = (w, \gamma)$, where $\gamma = \varphi$ or $\gamma = \eta$. By the properties of GFP-signatures $\mathrm{Sig}_\nu^{\chi(w)}(\gamma) \leq \mathrm{Sig}_\nu^{\chi(w)}(\vartheta)$. As the next node on the play we take $(w, \gamma)$.

*(iii)* If $\vartheta = \varphi \wedge \eta$ then we have $\mathrm{Sig}_\nu^{\chi(w)}(\gamma) \leq \mathrm{Sig}_\nu^{\chi(w)}(\vartheta)$ for $\gamma = \varphi$ or $\gamma = \eta$. As the next node on the play we take $(w, \gamma)$.

*(iv)* If $\vartheta = (\exists \boldsymbol{x} \,.\, \alpha(\boldsymbol{a}, \boldsymbol{x}))\eta(\boldsymbol{a}, \boldsymbol{x})$ then we have $f(w, \vartheta) = (w, \eta(\boldsymbol{a}, \boldsymbol{b}))$ for some $\boldsymbol{b}$ with $\alpha(\boldsymbol{a}, \boldsymbol{b}) \in \Gamma$, or $f(w, \vartheta) = (w', \vartheta)$ for some neighbour $w'$ of $w$ containing all the constants from $\boldsymbol{a}$. In the first case we have $\mathrm{Sig}_\nu^{\chi(w)}(\eta(\boldsymbol{a}, \boldsymbol{b})) \leq \mathrm{Sig}_\nu^{\chi(w)}(\vartheta)$ and we put $(w, \eta(\boldsymbol{a}, \boldsymbol{b}))$ as the next node on the path. Otherwise we put $(w', \vartheta)$ as the next node; cearly $\mathrm{Sig}_\nu^{\chi(w')}(\vartheta) = \mathrm{Sig}_\nu^{\chi(w)}(\vartheta)$.

*(v)* If $\vartheta = (\forall \boldsymbol{x} \,.\, \alpha(\boldsymbol{a}, \boldsymbol{x}))\eta(\boldsymbol{a}, \boldsymbol{x})$ then we must choose between $(w, \eta(\boldsymbol{a}, \boldsymbol{b}))$ for some $\boldsymbol{b}$ with $\alpha(\boldsymbol{a}, \boldsymbol{b}) \in \Gamma$, or $(w', \vartheta)$ for some neighbour $w'$ of $w$. We know that there is an extension $\chi'$ of $\chi(w)$ such that $\mathfrak{A}, \chi' \models \alpha(\boldsymbol{a}, \boldsymbol{x}) \wedge \neg\eta(\boldsymbol{a}, \boldsymbol{x})$ and moreover $\mathrm{Sig}_\nu^{\chi'}(\eta(\boldsymbol{a}, \boldsymbol{x})) \leq \mathrm{Sig}_\nu^{\chi'}(\vartheta)$. By Lemma 6.5 there is a node $v \in \bigcap_{a \in \boldsymbol{a}} \chi(a) \cap \bigcap_{x \in \boldsymbol{x}} \chi(x)$. If $v = w$ then we choose $(w, \eta(\boldsymbol{a}, \boldsymbol{b}))$ for $\boldsymbol{b}$ such that $\boldsymbol{x} = \chi(w)(\boldsymbol{b})$. If $v \neq w$ then we choose $(w', \vartheta)$ where $w'$ is a neighbour of $w$ which is on the path to $v$. As all the constants from $\boldsymbol{a}$ appear in the label of $w$ and in the label of $v$ they must also appear in the label of $w'$. Hence also $\vartheta$ appears in the label of $w'$.

*(vi)* In the remaining cases of fixpoint predicates and fixpoint variables there is nothing to choose.

The play constructed above is infinite. We have two cases. One is that a formula starting with a quantifier is never reduced. It cannot be an universal formula by the way the play is defined. If it is an existential formula then the play is winning for player 1.

The other case is that we have infinitely many positions where some variable $T_j$ is regenerated, i.e., $\gamma_i = T_j \boldsymbol{a}$ for infinitely many $i$. Let $h$ be the smallest among all the indices $j$ s.t. $T_j$ is regenerated infinitely often on the play. By the assumption that the strategy $f$ is winning for player 0, $T_h$ must be a GFP-variable. Let us look at the sequence of signatures $\{\mathrm{Sig}_\nu^{\chi(w_i)}(\vartheta_i)\}_{i=1,2,\ldots}$. By (the dual of) Lemma 3.6 from some point the signature never increases on positions $\leq h$. It also decreases on these positions each time $T_h$ is met. As this is impossible, $T_h$ must be an LFP-variable and the play is winning for player 1. □

**Corollary 6.7.** *Let $\psi$ be a well-named $\mu$LGF sentence of width $r$. The automaton $\mathcal{A}_\psi$, over a set of $2r$ constants, accepts some tableau for $\psi$ if and only $\psi$ is satisfiable.*

**Proof of Theorems 1.1 and 1.2** By results in [5], even the satisfiability problems for GF and LGF (without fixed points) are 2EXPTIME-hard.

Let $\psi \in \mu$LGF be a sentence of length $n$ and width $r$. Let $K$ be a set of constants of size $2r$. Note that if sentences are encoded over a fixed finite alphabet, then the encoding of a tuple of $r$ distinct variables has length $\Omega(r \log r)$, so $r \log r = \mathcal{O}(n)$. Hence the size of $cl(\psi, K)$ is $\mathcal{O}(n \cdot 2^{r \log r}) = 2^{\mathcal{O}(n)}$.

By Corollary 6.7 it is enough to check whether there exists a tableau for $\psi$ over constants $K$ that is accepted by $\mathcal{A}_\psi$. To check this we construct an automaton $\mathcal{B}_\psi$ recognizing all the tableaux for $\psi$ over $K$. The construction of $\mathcal{B}_\psi$ is routine; in particular, thanks to alternation, the states of $\mathcal{B}_\psi$ can be just the elements of $cl(\psi, K)$. Finally, let $\mathcal{C}_\psi$ be an automaton checking whether a given tree is accepted by both $\mathcal{A}_\psi$ and $\mathcal{B}_\psi$. (This construction is trivial for alternating automata.) We have that $\mathcal{C}_\psi$ accepts some tree iff $\psi$ is satisfiable. By Theorem 5.4 the emptiness of $\mathcal{C}_\psi$ can be decided in $\mathcal{O}(2^s)$ time, where $s$ is the number of states of $\mathcal{C}_\psi$. By the construction

$$s = \mathcal{O}(|cl(\psi, K)|) = \mathcal{O}(n \cdot 2^{r \log r}) = 2^{\mathcal{O}(n)}.$$

Hence the emptiness of $\mathcal{C}_\psi$ can be decided in time $2^{2^{\mathcal{O}(n)}}$. This gives the proof of Theorem 1.1. When $r$ is bounded by a constant we get $s = \mathcal{O}(n)$. This proves Theorem 1.2. □

## 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] F. DONNINI, M. LENZERINI, D. NARDI, AND A. SCHAERF, *Reasoning in description logics*, in Principles of Knowledge Representation, G. Brewka, ed., CSLI Publications, 1996, pp. 193–238.

[4] A. EMERSON AND C. JUTLA, *Tree automata, mu-calculus and determinacy*, in Proc. 32nd IEEE Symp. on Foundations of Computer Science, 1991, pp. 368–377.

[5] E. GRÄDEL, *On the restraining power of guards*, Journal of Symbolic Logic. To appear.

[6] E. GRÄDEL, P. KOLAITIS, AND M. VARDI, *On the decision problem for two-variable first-order logic*, The Bulletin of Symbolic Logic, 3 (1997), pp. 53–69.

[7] E. GRÄDEL AND M. OTTO, *On logics with two variables*, Theoretical Computer Science, (1999). To appear.

[8] E. GRÄDEL, M. OTTO, AND E. ROSEN, *Undecidability results on two-variable logics*, in Proceedings of 14th Symposium on Theoretical Aspects of Computer Science STACS'97, Lecture Notes in Computer Science Nr. 1200, Springer, 1997, pp. 249–260.

[9] E. GRÄDEL AND E. ROSEN, *On preservation theorems for two-variable logic*, Mathematical Logic Quarterly, (1999). To appear.

[10] M. MORTIMER, *On languages with two variables*, Zeitschrift für Mathematische Logik und Grundlagen der Mathematik, 21 (1975), pp. 135–140.

[11] A. MOSTOWSKI, *Games with forbidden positions*, Tech. Rep. Tech. Report 78, University of Gdansk, 1991.

[12] 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.

[13] C. STIRLING, *Bisimulation, model checking and other games*. Notes for the Mathfit instructional meeting on games and computation. Edinburgh, 1997.

[14] R. STREETT AND A. EMERSON, *An automata theoretic decision procedure for the propositional mu-calculus*, Information and Computation, 81 (1989), pp. 249–264.

[15] 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.

[16] ———, *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.

[17] W. ZIELONKA, *Infinite games on finitely coloured graphs with applications to automata on infinite trees*, Theoretical Computer Science, 200 (1998), pp. 135–183.