Information Processing Letters 112 (2012) 371–375

Contents lists available at SciVerse ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

# Finite satisfiability for guarded fixpoint logic

Vince Bárány [*], Mikołaj Bojańczyk [1]

*Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Banacha 2, 02-097 Warsaw, Poland*

ARTICLE INFO

*Article history:*
Received 21 February 2011
Received in revised form 8 February 2012
Accepted 8 February 2012
Available online 13 February 2012
Communicated by A. Muscholl

*Keywords:*
Formal methods
Guarded fragment
Guarded fixpoint logic
Finite satisfiability

ABSTRACT

The finite satisfiability problem for guarded fixpoint logic is decidable and complete for 2ExpTime (resp. ExpTime for formulas of bounded width).

## 1. Introduction

The *guarded fragment* (GF) is a robustly decidable syntactic fragment of first-order logic possessing many favourable model theoretic traits, such as the finite model property [5]. The guarded fragment has received much attention since its conception thirteen years ago [1] and has since seen a number of variants and extensions adopted in diverse fields of computer science. One of the most powerful extensions to date, *guarded fixpoint logic* ($\mu$GF) was introduced by Grädel and Walukiewicz in [6], who showed that the satisfiability problem of guarded fixpoint logic is computationally no more complex than for the guarded fragment: 2ExpTime-complete in general and ExpTime-complete for formulas of bounded width. Guarded fixpoint logic extends the modal $\mu$-calculus with backward modalities, hence it does *not* have the finite model property. Therefore, there is a finite satisfiability decision problem: to determine whether a formula has a finite model. Grädel and Walukiewicz left the decidability of this problem open. Here we claim this inheritance.

**Main Theorem 1.** *It is decidable whether or not a given guarded fixpoint sentence is finitely satisfiable. The problem is* 2ExpTime-*complete in general, and* ExpTime-*complete for formulas of bounded width.*

As noted above the stated hardness results already hold for the guarded fragment [5]. The proof of the upper bounds combines three ingredients:

(i) the tight connection between $\mu$GF and alternating automata [6];
(ii) decidability of emptiness of alternating automata over finite graphs [3];
(iii) a recent development in the finite model theory of guarded logics [2].

In what follows, no intricate knowledge of either [3] or [2] is required, the results of these papers are used as black boxes: (i) & (ii) provide the algorithm and the construction of (iii) proves its correctness. The stated time complexity results from combining those of (i) (Theorem 3 below) and (ii) (Theorem 2).

**Outline of the paper.** Guarded fixpoint logic and related notions are introduced in Section 2. In Section 3 we define alternating automata on undirected graphs, and state the

[*] Corresponding author.
*E-mail addresses:* vbarany@mimuw.edu.pl (V. Bárány), bojan@mimuw.edu.pl (M. Bojańczyk).

[1] Work supported by ERC Starting Grant "Sosna".

0020-0190/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.ipl.2012.02.005

result of [3]. Section 4 establishes the connection between guarded fixpoint logic and alternating automata along the lines of [6]. In Section 5, we present the algorithm and prove its correctness using [2].

## 2. Guarded fixpoint logic

The guarded fragment of first-order logic comprises only formulas with a restricted pattern of "guarded quantification" and otherwise inherits the semantics of first-order logic. Guarded quantification takes the form

$$\exists\bar{y}\big(R(\bar{x}\bar{y})\wedge\varphi(\bar{x}\bar{y})\big)\quad\text{or, dually,}\quad\forall\bar{y}\big(R(\bar{x}\bar{y})\to\varphi(\bar{x}\bar{y})\big)$$

where $R(\bar{x}\bar{y})$ is a positive literal acting as a *guard* by effectively restricting the variables $\bar{x}$ to range only over those tuples occurring in the appropriate positions in the atomic relation $R$. Here it is meant that $\bar{x}\bar{y}$ include all free variables of $\varphi$ in no particular order. A *guarded set* of elements of a relational structure $\mathfrak{A}$ is a set whose members occur among the components of a single relational atom $R(\bar{a})$ of $\mathfrak{A}$. Guarded quantification can be understood as a generalisation of polyadic modalities of modal logic. Indeed, the guarded fragment was conceived precisely with this analogy in mind [1], therefore it is no coincidence that the model theory of the guarded fragment bears such a strong resemblance to that of modal logic [7].

Guarded fixpoint logic is obtained by extending the guarded fragment of first-order logic with least and greatest fixpoint constructs. Its syntax can be defined by the following scheme

$$\varphi ::= R(\bar{x}) \mid \varphi\wedge\varphi' \mid \neg\varphi \mid \exists\bar{y}\big(R(\bar{x}\bar{y})\wedge\varphi''(\bar{x}\bar{y})\big) \mid Z(\bar{z}) \mid \\ \big[\mathsf{LFP}\,Z,\bar{z}\,.\,\varphi'''(Z,\bar{z})\big](\bar{x}) \mid \big[\mathsf{GFP}\,Z,\bar{z}\,.\,\varphi'''(Z,\bar{z})\big](\bar{x})$$

where $R$ is an arbitrary atomic relation symbol, $Z$ is a second-order fixpoint variable, where all free first-order variables of $\varphi''(\bar{x}\bar{y})$ and $\varphi'''(Z,\bar{z})$ are among those indicated, and $\varphi'''(Z,\bar{z})$ is required to be positive in $Z$. The semantics is standard: the least (or greatest) fixpoint of a formula $\varphi'''(Z,\bar{z})$ on a given structure is the wrt. set inclusion least (resp. greatest) relation $S$ satisfying $S(\bar{a})\leftrightarrow\varphi'''(S,\bar{a})$ for all $\bar{a}$ on the structure. Crucially, fixpoint variables and fixpoint formulas are not allowed to stand as guard in a guarded quantification, only atomic relation symbols may act as guards. Furthermore, within sentences it can be assumed wlog. that in the matrix $\varphi'''(Z,\bar{z})$ of a fixpoint formula the tuple of free variables $\bar{z}$ is explicitly guarded [6].

Guarded fixpoint logic naturally extends the modal $\mu$-calculus with backward modalities. As such it can axiomatise (the necessarily infinite) well-founded directed acyclic graphs having no sink nodes, e.g. as follows.

$$\exists xy\, E(x,y)\wedge\forall xy\big(E(x,y) \\ \to\big[\mathsf{LFP}\,Z,z\,.\,\forall v E(v,z)\to Z(v)\big](x)\wedge\exists w E(y,w)\big)$$

*Guarded bisimulation.* Guarded logics possess a very appealing model theory in which guarded bisimulation plays a similarly central role as does bisimulation for modal logics. A *guarded bisimulation* [1,7] between two structures $\mathfrak{A}_0$ and $\mathfrak{A}_1$ of the same relational signature is a family $Z$ of partial isomorphisms $\alpha: A_0\to A_1$ with $A_i\subseteq\mathfrak{A}_i$, satisfying the following back-and-forth conditions. (i) For every $\alpha: A_0\to A_1$ in $Z$ and every *guarded* subset $B_0$ of $\mathfrak{A}_0$ there is a partial isomorphism $\gamma: C_0\to C_1$ in $Z$ with $B_0\subseteq C_0$ and $\alpha(x)=\gamma(x)$ for all $x\in A_0\cap C_0$. (ii) For every $\alpha: A_0\to A_1$ in $Z$ and every *guarded* subset $B_1$ of $\mathfrak{A}_1$ there is a partial isomorphism $\gamma: C_0\to C_1$ in $Z$ with $B_1\subseteq C_1$ and $\alpha^{-1}(y)=\gamma^{-1}(y)$ for all $y\in A_1\cap C_1$. We write $\mathfrak{A}_0,\bar{a}\sim_{\mathrm{g}}\mathfrak{A}_1,\bar{b}$ to signify that there is a guarded bisimulation $Z$ between $\mathfrak{A}_0$ and $\mathfrak{A}_1$ with $(\bar{a}\mapsto\bar{b})\in Z$ and say that $\bar{a}$ of $\mathfrak{A}_0$ and $\bar{b}$ of $\mathfrak{A}_1$ are *guarded bisimilar*.

Guarded bisimilarity is an equivalence relation on the set of guarded tuples of any relational structure, and guarded fixpoint formulas are invariant under guarded bisimulation [7]: if $\mathfrak{A},\bar{a}\sim_{\mathrm{g}}\mathfrak{B},\bar{b}$ then for every guarded fixpoint formula $\varphi$ it holds that $\mathfrak{A}\models\varphi(\bar{a})$ iff $\mathfrak{B}\models\varphi(\bar{b})$. The guarded fragment has been characterised as the guarded-bisimulation-invariant fragment of first-order logic, most recently even in the context of finite structures [8]. Similarly, guarded fixpoint logic is characterised as the guarded-bisimulation-invariant fragment of guarded second-order logic [7].

## 3. Alternating two-way automata

In this section, we introduce alternating automata on undirected graphs. A similar model, namely alternating two-way automata on infinite trees, was used by Grädel and Walukiewicz [6] in their decision procedure for satisfiability of guarded fixpoint logic. They reduced satisfiability to the emptiness problem for alternating two-way automata on infinite trees. The latter problem was shown to be decidable by Vardi [9].

In [9,3,4] a two-way automaton navigating an infinite tree has the choice of moving its head either to the parent or to a child node, or staying in its current location. In this paper, instead of automata on directed trees, we consider automata on undirected graphs. In an undirected graph, the automaton can only choose to stay in place or to move to a neighbouring vertex. This is in the spirit of [6], where automata on directed trees were employed, which did not actually distinguish between parent and child nodes.

An *alternating automaton on undirected graphs* is defined by: an input alphabet $\Sigma$, a finite set of states $Q=Q_\forall\uplus Q_\exists$, an initial state $q_I$, a ranking function $\Omega: Q\to\mathbb{N}$ for the parity acceptance condition, and a transition relation

$$\delta\subseteq Q\times\Sigma\times\{\mathsf{stay},\mathsf{move}\}\times Q\,.$$

An input to the automaton is an undirected graph whose nodes are labelled by $\Sigma$, and a designated node $v_0$ of the graph. The automaton accepts an input graph $G$ from an initial node $v_0$ if player $\exists$ wins the parity game defined below.

The arena of the parity game consists of pairs of the form $(v,q)$, where $v$ is a node of $G$, and $q$ is a state of the automaton. The initial position in the arena is $(v_0,q_I)$. The rank of a position $(v,q)$, as used by the parity condition, is $\Omega(q)$. Let $u$ be a node of the input graph, and let $a\in\Sigma$ be its label. In the arena of the game, there is an edge from $(u,q)$ to $(w,p)$ if:

- there is a transition $(q, a, \mathsf{stay}, p)$ and $u = w$; or
- there is a transition $(q, a, \mathsf{move}, p)$ and $\{u, w\} \in E(G)$.

Some alternating automata on undirected graphs accept only infinite graphs. (Given a 3-colouring of a graph by $\{0, 1, 2\}$, edges can be directed so that 'target colour' – 'source colour' $\equiv 1 \bmod 3$. An automaton can verify 3-colouring and well-foundedness of the induced digraph and check for an infinite forward path.) Therefore, it makes sense to ask: does a given automaton accept some finite graph? This problem was shown decidable in [3,4].

**Theorem 2.** *(See [3,4].) Given an alternating automaton on undirected graphs it is decidable in exponential time in the number of states of the automaton, whether or not it accepts some finite graph.*

Formally, [3,4] considered two-way automata on directed graphs with the automaton having transitions corresponding to: staying in the same node, moving forward along an edge, and moving backward along an edge. Clearly, the two-way model is more general than the one for undirected graphs.

*Undirected bisimulation.* We write nodes$(G)$ for the nodes of a graph $G$. Consider two undirected graphs $G_0$ and $G_1$, with node labels. An undirected bisimulation is a set

$$Z \subseteq \mathrm{nodes}(G_0) \times \mathrm{nodes}(G_1)$$

with the following properties. If $(v_0, v_1)$ belongs to $Z$, then the node labels of $v_0$ and $v_1$ are the same. Also, for any $i \in \{0, 1\}$ and node $w_i$ connected to $v_i$ by an edge, there exits a node $w_{1-i}$ connected to $v_{1-i}$ by an edge and such that $(w_0, w_1) \in Z$. We say that node $v_0$ of a graph $G_0$ is bisimilar to node $v_1$ in a graph $G_1$ if there is an undirected bisimulation that contains the pair $(v_0, v_1)$. In this case, for every alternating automaton on undirected graphs, the automaton accepts $G_0$ from $v_0$ if and only if it accepts $G_1$ from $v_1$.

*Undirected unravelling.* Consider an undirected graph $G$ and $v$ a node of $G$. The undirected unravelling of $G$ from $v$ is the graph $T$, whose nodes are paths in $G$ that begin in $v$, and edges are placed between a path and the same path without the last node. The undirected unravelling is a tree. We write

$$\pi : \mathrm{nodes}(T) \to \mathrm{nodes}(G)$$

for the function that maps a path to its terminal node. If $G$ has node labels, then one labels the nodes of $T$ according to their images under $\pi$. Then, the graph of $\pi$ is an undirected bisimulation between $T$ and $G$.

## 4. Tabloids

Below we work with undirected graphs representing templates of relational structures. We call them tabloids alluding to their semblance to the tableaux of [6]. Tabloids are also reminiscent of the 'guarded bisimulation invariants' of [2]. Intuitively, vertices of a tabloid represent templates for guarded substructures and edges signify their overlap. The precise manner of overlap is implicitly coded by repeated use of constant names appearing in vertex labels. By contrast, [2,7] code overlaps explicitly as edge labels.

*Tabloid.* Fix a relational signature $\Sigma$ and a set $K$ of constant names. A *tabloid* over signature $\Sigma$ and constants $K$ is an undirected graph, where every node $v$ is equipped with two labels: a set $K_v \subseteq K$, called the *constants of* $v$, and an atomic $\Sigma$-type $\tau_v$ over $K_v$, called the *type of* $v$. If nodes $v$ and $w$ are connected by an edge in the graph, then the types $\tau_v$ and $\tau_w$ should agree over the constants from $K_v \cap K_w$.

*A structure from a tree tabloid.* Consider a tabloid $T$ whose underlying graph is a tree. We define a $\Sigma$-structure $\mathfrak{A}(T)$ as follows. The universe of $\mathfrak{A}(T)$ is built using pairs $(v, c)$, where $v$ is a vertex of $T$ and $c$ is a constant of $v$. The universe consists not of these pairs, but of their equivalence classes under the following equivalence relation: $(v, c)$ and $(v', c')$ are equivalent if $c = c'$ and $c$ occurs in the label of every node on the undirected path connecting $v$ and $v'$ in $T$. The path is unique, because the underlying graph is a tree. We write $[v, c]$ for an equivalence class of such a pair. A tuple $([v_1, c_1], \ldots, [v_n, c_n])$ satisfies a relation $R \in \Sigma$ in $\mathfrak{A}(T)$ if there is some node $v$ such that

$$[v, c_1] = [v_1, c_1], \ldots, [v, c_n] = [v_n, c_n] \tag{1}$$

and $R(c_1, \ldots, c_n)$ is implied by $\tau_v$. Because $T$ is a tree, this definition does not depend on the choice of $v$, since the set of nodes $v$ satisfying (1) is connected. It is, however, unclear how to extend this construction to cyclic tabloids.

*Labelling with a formula.* Consider a tree tabloid $T$ over constants $K$ and signature $\Sigma$. Let $\varphi$ be a formula over $\Sigma$. Consider a node $v$ of $T$ with constants $K_v$, a subformula $\psi$ of $\varphi$, and a function $\eta$ that maps free variables of $\psi$ to constants in $K_v$. For $v$ and $\eta$, define a valuation $[\eta]_v$, which maps free variables of $\psi$ to elements of the structure $\mathfrak{A}(T)$, by setting $[\eta]_v(x) = [v, \eta(x)]$.

The *$\varphi$-type* of the node $v$ is the set of pairs $(\psi, \eta)$ such that $\psi$ is a subformula of $\varphi$ or a literal in the signature of $\varphi$, and such that $\psi$ is valid in $\mathfrak{A}(T)$ under the valuation $[\eta]_v$. Thus each $\varphi$-type determines a unique atomic type. The set of $\varphi$-types is finite and depends on $K$ and $\varphi$ alone, call this set $\Gamma_{\varphi,K}$. Given a tree tabloid $T$ and $\varphi$, we define $T_\varphi$ to be the tree with the same nodes and edges as $T$, but where every node is labelled by its $\varphi$-type.

Recall that the width of a formula is the maximal number of free variables in any of its subformulas. The following was established in [6].

**Theorem 3.** *(See [6].) Let $\varphi$ be a guarded fixpoint sentence of width $n$ and let $K$ be a set of $2n$ constants. One can compute an alternating automaton $\mathcal{A}_\varphi$ on $\Gamma_{\varphi,K}$-labelled undirected graphs, such that $\mathcal{A}_\varphi$ accepts a tree $\Upsilon$ if and only if*

*$\Upsilon$ is of the form $T_\varphi$ for a tree tabloid $T$ such that $\mathfrak{A}(T) \models \varphi$*

*The number of states of $\mathcal{A}_\varphi$, and the time to compute it, are $O(|\varphi| \cdot \exp(n))$.*

## 5. Algorithm for finite satisfiability

We now propose the algorithm for finite satisfiability of guarded fixpoint logic. Given a formula $\varphi$, we compute the automaton $\mathcal{A}_\varphi$ using Theorem 3. Then, we test if the automaton $\mathcal{A}_\varphi$ accepts some finite graph, using Theorem 2. The combined running time clearly meets the claim of Theorem 1. This section is devoted to proving the correctness of this procedure.

**Proposition 4.** *A formula $\varphi$ of guarded fixpoint logic has a finite model if, and only if, the associated automaton $\mathcal{A}_\varphi$ accepts a finite graph.*

### 5.1. From a finite accepted graph to a finite model

First we prove that if the automaton $\mathcal{A}_\varphi$ accepts a finite graph $G_\varphi$, then $\varphi$ is satisfied in some finite structure. By Theorem 3, the undirected unravelling of $G_\varphi$, equally accepted by $\mathcal{A}_\varphi$, takes the form $T_\varphi$ for a tree tabloid $T$ such that $\mathfrak{A}(T) \models \varphi$. In fact, $T$ is the undirected unravelling of the finite tabloid $G$ obtained from $G_\varphi$ by restricting its labels to atomic types.

**Lemma 5.** *Let $G$ be a finite tabloid and $T$ its undirected unravelling. Then $\sim_{\mathrm{g}}$ has finite index on the set of guarded tuples of $\mathfrak{A}(T)$.*

**Proof.** All guarded subsets of $\mathfrak{A}(T)$ are of the form $\{[v, c_1], \ldots, [v, c_r]\}$ where $c_1, \ldots, c_r \in K$ are constant names appearing in the label of $v \in \mathrm{nodes}(T)$. Let $\pi : nodes(T) \to nodes(G)$ be the natural projection from $T$ onto $G$. Then $(T, v) \cong (T, w)$ whenever $\pi(v) = \pi(w)$, so it suffices to show the following.

**Claim 6.** $\mathfrak{A}(T), ([v, c_1], \ldots, [v, c_r]) \sim_{\mathrm{g}} \mathfrak{A}(T), ([w, c_1], \ldots, [w, c_r])$ *for every $v$ and $w$ such that $(T, v) \cong (T, w)$ and $\{c_1, \ldots, c_r\} = K_v = K_w$.*

Let for each $v$ and $w$ as in the claim $\alpha_{v,w}$ be the partial function mapping $[v, c] \mapsto [w, c]$ for all $c \in K_v$. By definition of $\mathfrak{A}(T)$ we have that each $\alpha_{v,w}$ is a partial isomorphism among guarded subsets of $\mathfrak{A}(T)$. We claim that

$$Z = \left\{\alpha_{v,w} \;\middle|\; (T, v) \cong (T, w)\right\}$$

is a guarded bisimulation. Take any $\alpha_{v,w} \in Z$ and guarded subset $B$ of $\mathfrak{A}(T)$. Then $B = \{[u, d_1], \ldots, [u, d_s]\}$ for some $u \in \mathrm{nodes}(T)$ and constant names $D = \{d_1, \ldots, d_s\} \subseteq K_u$. Because $(T, v) \cong (T, w)$ there is a $y \in \mathrm{nodes}(T)$ such that $(T, v, u) \cong (T, w, y)$. In particular, $B \subseteq \mathrm{dom}(\alpha_{u,y})$, and the paths connecting $v$ with $u$ and $w$ with $y$ are isomorphic. We thus have for every $i \leqslant r$ and $j \leqslant s$ that $[v, c_i] = [u, d_j]$ iff $c_i = d_j$ and $c_i \in K_z$ for every node $z$ on the path connecting $v$ and $u$ (equivalently, on the path connecting $w$ and $y$) iff $[w, c_i] = [y, d_j]$. Therefore, $\alpha_{u,y}$ and $\alpha_{v,w}$ agree on $\mathrm{dom}(\alpha_{u,y}) \cap \mathrm{dom}(\alpha_{v,w})$, and $\alpha_{u,y}^{-1}$ and $\alpha_{v,w}^{-1}$ agree on $\mathrm{rng}(\alpha_{u,y}) \cap \mathrm{rng}(\alpha_{v,w})$. This shows that $Z$ satisfies the 'forth property' and, by symmetry, also the 'back property', as needed. □

Note that, in stark contrast to bisimulation on graphs, there is no apparent way of defining a quotient $\mathfrak{A}(T)/{\sim_{\mathrm{g}}}$. Nevertheless, we can obtain a finite structure guarded bisimilar to $\mathfrak{A}(T)$ using the following result.

**Theorem 7.** *(See [2, Theorem 6], cf. also [8].) Every relational structure on which $\sim_{\mathrm{g}}$ has finite index is guarded bisimilar to a finite structure.*

### 5.2. From a finite model to a finite accepted graph

Next we prove that if $\varphi$ has a finite model then $\mathcal{A}_\varphi$ of Theorem 3 accepts some finite graph. Recall that all graphs accepted by $\mathcal{A}_\varphi$ are labelled by $\varphi$-types from $\Gamma_{\varphi,K}$, where $K$ is a set of $2n$ constants, with $n$ the width of $\varphi$. So let $\mathfrak{A}$ be a finite model of $\varphi$. Wlog. all guarded subsets of $\mathfrak{A}$ are of size at most $n$ (as $\varphi$ is oblivious to relational atoms with more than $n$ distinct components, these can be removed from $\mathfrak{A}$).

We define a finite tabloid $G$ as follows. Vertices of $G$ are injections $\chi : A \to K$, where $A$ is a guarded subset of $\mathfrak{A}$. For each vertex $\chi$ its set of constants is $K_\chi = \mathrm{rng}(\chi)$, and its type $\tau_\chi$ is the image of the atomic type of $A$ in $\mathfrak{A}$ under $\chi$. Two vertices $\chi$ and $\chi'$ are adjacent in $G$ just if $\chi \cup \chi'$ is an injective function. This ensures that adjacent nodes are labelled with consistent types, i.e. that $G$ is indeed a tabloid.

Let $T$ be the undirected unravelling of $G$, and $\pi : \mathrm{nodes}(T) \to \mathrm{nodes}(G)$ the natural projection. Then $(T, v) \cong (T, w)$ whenever $\pi(v) = \pi(w)$. From Claim 6 and the guarded bisimulation invariance of $\mu$GF it follows that $v$ and $w$ have the same label in $T_\varphi$ whenever $\pi(v) = \pi(w)$. Hence, it make sense to define $G_\varphi$ as having the same underlying graph as $G$ with each $\chi \in \mathrm{nodes}(G)$ labelled exactly as any and all nodes in $\pi^{-1}(\chi)$. Then $T_\varphi$ is isomorphic to the undirected unravelling of $G_\varphi$. By Theorem 3, $\mathcal{A}_\varphi$ accepts $G_\varphi$ iff it accepts $T_\varphi$ iff $\mathfrak{A}(T) \models \varphi$. Thus, to conclude, it suffices to prove the following.

**Claim 8.** $\mathfrak{A} \sim_{\mathrm{g}} \mathfrak{A}(T)$.

**Proof.** For each $v \in \mathrm{nodes}(T)$, $\pi(v)$ is an injection $\chi_v : A_v \to K_v$ from a guarded subset $A_v$ of $\mathfrak{A}$ to the set $K_v$ of constant names in the label of $v$. Let $\gamma_v : K_v \to \mathfrak{A}(T)$ map each $c \in K_v$ to $[v, c]$. Then $\gamma_v \circ \chi_v$ is a partial isomorphism between guarded subsets of $\mathfrak{A}$ and $\mathfrak{A}(T)$. We claim that $\{\gamma_v \circ \chi_v \mid v \in \mathrm{nodes}(T)\}$ is a guarded bisimulation between $\mathfrak{A}$ and $\mathfrak{A}(T)$.

*'Forth':* Consider $\gamma_v \circ \chi_v : A_v \to \{[v, c] \mid c \in K_v\}$ and $B$ a guarded subset of $\mathfrak{A}$. Then, since $|B \cup A| \leqslant |K| = 2n$, there is a vertex $\chi : B \to K$ such that $\chi_v|_{A_v \cap B} = \chi|_{A_v \cap B}$ and $\chi(A_v) \cap \chi'(B) = \chi(A_v \cap B)$. It follows that $\chi$ is adjacent to $\chi_v$ in $G$, hence $w = v \cdot \chi$ is adjacent to $v$ in $T$,

$\pi(w) = \chi_w = \chi$, and that thus $\gamma_w \circ \chi_w$ fulfills the requirements of the 'forth property'.

*'Back':* Consider now $\gamma_v \circ \chi_v : A_v \to \{[v,c] \mid c \in K_v\}$ and a guarded subset $B = \{[w,d] \mid d \in D\}$ of $\mathfrak{A}(T)$. Let $C = D \cap K_v$. The intersection of $B$ and $\{[v,c] \mid c \in K_v\}$ consists of those $[v,c]$ such that $c \in C$ appears in the label of every node along the path $\rho$ connecting $v$ to $w$ in $T$. Let $u$ and $y$ be adjacent nodes of $\rho$. Then $\pi(u) = \chi_u$ and $\pi(y) = \chi_y$ are adjacent in $G$ and thus ${\chi_u^{-1}}|C = {\chi_y^{-1}}|C$. By induction we get that ${\chi_v^{-1}}|C = \chi_w^{-1}|C$. It follows that $\gamma_w \circ \chi_w$ satisfies the requirements of the 'back property'. □

This completes the proof of Proposition 4, thereby also our Main Theorem 1.

## Acknowledgements

We gratefully acknowledge the invaluable help the reviewers have provided in improving this paper.

## References

[1] H. Andréka, J. van Benthem, I. Németi, Modal languages and bounded fragments of predicate logic, Journal of Philosophical Logic 27 (1998) 217–274.

[2] V. Bárány, G. Gottlob, M. Otto, Querying the guarded fragment, in: Proc. LICS'10, IEEE Computer Society, 2010, pp. 1–10.

[3] M. Bojańczyk, Two-way alternating automata and finite models, in: Proc. ICALP'02, in: Lecture Notes in Computer Science, vol. 2380, Springer, 2002, pp. 833–844.

[4] M. Bojańczyk, Decidable properties of tree languages, PhD Thesis, University of Warsaw, 2004.

[5] E. Grädel, On the restraining power of guards, Journal of Symbolic Logic 64 (4) (1999) 1719–1742.

[6] E. Grädel, I. Walukiewicz, Guarded fixed point logic, in: Proc. LICS'99, 1999, pp. 45–54.

[7] E. Grädel, C. Hirsch, M. Otto, Back and forth between guarded and modal logics, ACM Transactions on Computational Logics 3 (3) (2002) 418–463.

[8] M. Otto, Highly acyclic groups, hypergraph covers and the guarded fragment, in: Proc. LICS'10, IEEE Computer Society, 2010, pp. 11–20.

[9] M. Vardi, Reasoning about the past with two-way automata, in: Proc. LICS'10, in: Lecture Notes in Computer Science, vol. 1443, 1998, pp. 628–641.