Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®

Theoretical Computer Science 298 (2003) 511 – 528

Theoretical Computer Science

www.elsevier.com/locate/tcs

The finite graph problem for two-way alternating automata

Mikołaj Bojańczyk[1]

Uniwersytet Warszawski, Wydział MIM, Banacha 2, Warszawa, Poland

Abstract

Two-way alternating automata on infinite trees were introduced by Vardi (Reasoning about the part with two way automata, Lecture Notes in Computer Science, vol. 11, Springer, Berlin, 1998, pp. 628–641). Here we consider alternating two-way automata on graphs and show the decidability of the following problem: "does a given automaton with the Büchi condition accept any *finite* graph?" Using this result we demonstrate the decidability of the finite model problem for a certain fragment of the modal μ-calculus with backward modalities.

Keywords: Alternating automata; Finite model; μ-calculus

1. Introduction

Alternating tree automata with the parity condition were introduced by Emmerson and Jutla in [2]. In terms of expressibility these automata are nothing new—they define the same class of languages as the simpler nondeterministic tree automata on the one hand and as the powerful monadic second order theory on trees (S2S) on the other. Nevertheless, the formalism of alternating automata offers a good balance between logical manageability and computational complexity. Emptiness for alternating tree automata can be tested in EXPTIME, which is far better than the non-elementary procedures for S2S; on the other hand, closure under negation is trivial for alternating automata and very difficult for nondeterministic automata (cf. the famous "complementation lemma" [11]).

[1] Partly supported by KBN Grant 7 T11C 027 20.
E-mail address: bojan@mimuw.edu.pl (M. Bojańczyk).

doi:10.1016/S0304-3975(02)00866-6

Alternating automata are also very closely connected the modal μ-calculus of Kozen [7]. The two formalisms are so similar, that one can actually view one as syntactic sugar for the other. For exactly the same reason as alternating automata, the μ-calculus is a very good program logic and is used in program verification and analysis. The μ-calculus context is the reason why *two-way* alternating automata on infinite trees were introduced by Vardi. In [14], the satisfiability problem for the propositional μ-calculus with backward modalities was proven decidable via a reduction to two-way alternating automata.

In the μ-calculus with *backwards modalities*, besides the usual least and greatest fix-point operators μ, ν and modal quantification $\exists^{+1}$, $\forall^{+1}$ (sometimes written as $\diamond$, $\Box$), of the one-way propositional μ-calculus, one allows for quantification over backward modalities, denoted by $\exists^{-1}$ and $\forall^{-1}$. A formula of the form $\exists^{-1}\phi$ states that ϕ occurs in some predecessor of the current state, similarly for $\forall^{-1}$. Analogously to the μ-calculus, a two-way automaton can, apart from the usual forward moves of one-way alternating automata, make backward moves.

In [4], the two-way alternating automaton was used to solve another satisfiability problem—this time for Guarded Fixed Point Logic, an extension of the Guarded Fragment by fix-point operators. The Guarded Fragment (GF) is a subset of first order logic where quantification is restricted to so-called *guarded* quantifiers of the form

$$\forall \mathbf{y}.\ R(\mathbf{x},\mathbf{y}) \Rightarrow \phi(\mathbf{x},\mathbf{y}) \quad \exists \mathbf{y}.\ R(\mathbf{x},\mathbf{y}) \wedge \phi(\mathbf{x},\mathbf{y})$$

for tuples of variables $\mathbf{x}$, $\mathbf{y}$. The relation symbol R is called the *guard* of the formula. GF was originally introduced in [4] as an elaboration on the translation of modal logic into first order logic and is currently subject to much research.

There is an interesting common denominator in Guarded Fixed Point Logic, the μ-calculus with backward modalities and two-way alternating automata: none of them have a finite model property. We say a logic has the *finite model* property if every satisfiable sentence is satisfiable in some finite structure. The fact that adding past modalities to modal logic results in losing the finite model property can be traced back to [10]. Modal logic and even the modal μ-calculus have the finite model property; the μ-calculus with backward modalities does not (consider the sentence $\nu X.\exists^{+1}X \wedge \mu Y. \forall^{-1}Y$). A similar situation occurs in the Guarded Fragment: the fix-point extension no longer has the finite model property, contrary to the "bare" Guarded Fragment [3] and some of its other extensions (most notably the loosely Guarded Fragment [6]). For alternating two-way automata, a suitable example is presented in Section 2.

These observations give rise to the following decision problem: "Is a given sentence of the modal μ-calculus with backward modalities (or guarded fixed point logic) satisfiable in some *finite* structure?" While tackling this problem, we took the automata approach. However, for reasons sketched below, it turns out that we need a new definition of two-way alternating automata.

Most modal logics have a bisimulation-invariance property and the two-way μ-calculus is no exception. In particular, a sentence of the two-way μ-calculus cannot distinguish between a Kripke structure and its tree unraveling, and so every satisfiable sentence is satisfiable in a tree-like structure. Thus for the purpose of deciding satisfiability, one can constrain attention to tree models. This was the approach taken by Vardi; in fact his automata were alternating two-way automata on *infinite trees*.

As much as the tree model property is helpful in analyzing the satisfiability problem, things get more complicated where the finite model problem is concerned. The reason is that, unfortunately, finite models rarely turn out to be trees. There are finitely satisfiable sentences that have no finite tree models, for instance $\nu X.\exists^{+1}X$. For this reason, while investigating the finite model problem we will consider automata on arbitrary graphs, not on trees. Using a close correspondence between such automata and the μ-calculus with backward modalities, we will reduce the finite model problem to the *finite graph* problem for automata.

In this paper we consider alternating two-way automata with the Büchi acceptance condition. The Büchi acceptance condition means the automaton has a special, *accepting*, subset of the set of all states and in order for a run of the automaton to be accepting, one of the accepting states must occur infinitely often on every computation path. Even though it is weaker than the full parity condition of normal two-way automata, the Büchi condition is sufficient to recognize a large class of graph languages. The main result of this paper is a proof of the decidability of the finite graph problem for two-way alternating automata with the Büchi acceptance condition. Having done this, we decide the finite model property for a certain subset of the μ-calculus with backward modalities by a reduction to the finite graph problem for alternating two-way Büchi automata.

For the decidability proof of the finite graph problem, given an alternating two-way automaton $\mathscr{A}$, we wish to construct a nondeterministic automaton $\mathscr{A}'$ on trees that accepts some unravelings of finite graphs accepted by $\mathscr{A}$. In order to find a way of distinguishing unravelings of finite and infinite graphs, we introduce the concept of a graph *signature*. A two-way alternating automaton's signature in a particular vertex of a graph says what is the length of the longest sequence of non-accepting states that can appear in a run of the automaton beginning with that vertex. It can be proven that finite graphs have finite signatures, moreover—and this is the key property—unravelings of finite graphs also have finite signatures. In a sense, the reverse implication also holds: it turns out that accepting a tree, perhaps infinite, of finite signature is sufficient for accepting a finite graph; we discover one can "loop" a finite signature tree into an acceptable finite graph.

The essential technical "small signature" theorem allows us to find a tractable bound on the signature of finite graphs. The proof of this theorem uses a new approach of successive tree approximations of the final small signature tree. Using this theorem, the aforementioned nondeterministic automaton $\mathscr{A}'$ accepts trees where the signature of the automaton $\mathscr{A}$ is bounded by the constant in the small signature theorem.

In the last section of the paper we introduce the μ-calculus with backward modalities and, closely following a paper by Vardi [14], show its correspondence with alternating automata with the parity condition. We then show that the ν-μ fragment corresponds to Büchi condition automata and prove the decidability of the finite model problem for this fragment.

2. Two-way alternating automata

2.1. Games with the parity condition

In the eighties and nineties, games with the parity condition emerged as a powerful tool in the field of infinite tree automata [5,8,2]. A game point of view can simplify heretofore complicated matters; for instance the simplest known proofs of Rabin's Theorem use alternating automata and parity condition games. Thus we find it best to define the semantics of our two-way alternating automata using a game approach.

In this section we briefly define games with the parity condition and quote the key memoryless determinacy theorem. A more detailed exposition can be found in [15].

Definition 2.1 (*Parity condition game*). A *game with the parity condition* is a tuple $\mathbb{G} = \langle V_0, V_1, E, v_0, \Omega \rangle$, where V_0 and V_1 are disjoint countable sets of *positions*, the function $\Omega : V = V_0 \cup V_1 \to \{0, \dots, N\}$ is called the *coloring* function, $E \subseteq V \times V$ is the set of *edges*, and $v_0 \in V$ is some fixed *starting position*.

The game is played as follows. The play starts in the vertex v_0. Assuming the play has reached in turn j a vertex $v_j \in V_i$, $i \in \{0, 1\}$, the player i chooses some vertex v_{j+1} such that $(v_j, v_{j+1}) \in E$. If at some point one of the players cannot make a move, she loses. Otherwise, the winner depends on the infinite sequence $v_0, v_1, \dots$ of vertices visited in the game. This infinite play is winning for player 0 if the sequence $\Omega(v_0), \Omega(v_1), \dots$ satisfies the parity condition defined below, otherwise it is winning for player 1.

Definition 2.2 (*Parity condition*). A sequence $\{a_i\}$ of numbers belonging to some finite set of natural numbers is said to satisfy the *parity condition* if the smallest number occurring infinitely often in $\{a_i\}_{i \in \mathbb{N}}$ is even.

A strategy for the player $i \in \{0, 1\}$ is a mapping $s : V^* V_i \to V$ such that for each $v_0 \dots, v_j \in V^* V_i$, there is an edge in E from v_j to $s(v_0 \dots, v_j)$. We say a strategy is *memoryless* if $s(v_0 \dots, v_j)$ depends solely upon v_j. The concept of winning strategy is defined in the usual way. A very important theorem [2,8], which will enable us to consider only memoryless strategies, says the following:

Theorem 2.1 (Memoryless determinacy theorem). *Every game with the parity condition is determined, i.e. one of the players has a winning strategy. Moreover, the winner also has a memoryless winning strategy.*

2.2. The automaton

Two-way alternating automata were introduced by Vardi in [14] as a tool for deciding the satisfiability problem of the modal μ-calculus with backward modalities. As opposed to "normal" alternating automata, two-way automata can travel backwards across vertices. For purposes of the finite model problem, we consider a graph version of the automata.

Given a set of states Q, we consider formulas $\mathrm{Form}(Q)$ built using the logical connectives $\vee$ and $\wedge$ from atoms of the form $\forall Q$ and $\exists Q$. We define $\forall Q$ as the set of atoms of the form $\forall^{+1}q, \forall^{-1}q$ where $q \in Q\}$ and, similarly, $\exists Q$ as atoms of the form $\exists^{+1}q, \exists^{-1}q$. Moreover, we partition the set $\mathrm{Form}(Q)$ into *conjunctive* formulas $\mathrm{Con}(Q)$, i.e. either atoms from $\forall Q$ or formulas of the form $\phi_1 \wedge \phi_2$ and *disjunctive* formulas $\mathrm{Dis}(Q)$, i.e. atoms from $\exists Q$ and formulas of the form $\phi_1 \vee \phi_2$, where $\phi_1, \phi_2 \in \mathrm{Form}(Q)$.

Definition 2.3 (*Two-way alternating automaton*). A *two-way alternating automaton* on Σ-labeled graphs is a tuple of the form:

$$\langle Q, q_0, \Sigma, \delta, F \rangle$$

Q is a finite set of *states*, $q_0 \in Q$ is called the *starting state* and $F \subseteq Q$ is the set of *accepting states*. The *transition function* δ is of the form $\delta : Q \times \Sigma \to \mathrm{Form}(Q)$.

In this paper, when speaking of graphs, we will use *labeled graphs with a starting position*. Such a graph is a tuple $G = \langle V, E, \Sigma, e, v_0 \rangle$, where V is the set of *vertices*, $E \subseteq V \times V$ is the set of *edges*, the *labeling* is a function $e : V \to \Sigma$ and $v_0 \in V$ is the *starting position*. We assume the set Σ of *labels* is finite.

We write $\overset{\exists}{\forall}$ to denote any one of the quantifiers $\exists$ and $\forall$. For any edge $(u, w) \in E$ we will write $(u, w)^{-1}$ to denote the reverse edge, that is (w, u). To further simplify notation, assume for $(u, w)^1$ the edge (u, w).

To define the semantics of two-way alternating automata, we shall use games with the parity condition. Given a labeled graph $G = \langle V, E, \Sigma, e, v_0 \rangle$ and a two-way alternating automaton $\mathscr{A} = \langle Q, q_0, \Sigma, \delta, F \rangle$, we define below the game $\mathbb{G}(\mathscr{A}, G) = \langle V_0, V_1, E', v_0', \Omega \rangle$.

The set of game positions of $\mathbb{G}(\mathscr{A}, G)$ is defined $V_0 = \mathrm{Dis}(Q) \times V$ and $V_1 = \mathrm{Con}(Q) \times V$. Game positions from the set $(\forall Q \cup \exists Q) \times V$ are called *atomic game positions*. The edges of the game are set as follows:

- For atomic game positions $(\overset{\exists}{\forall}{}^i q, v)$, we place in $\mathbb{G}(\mathscr{A}, G)$ an edge to $(\delta(q, e(w)), w)$ if $(v, w)^i \in E$.
- For a non-atomic game positions (ϕ, v) there exists an edge to the game position (ϕ', v) for each immediate sub-formula ϕ' of ϕ.

The coloring Ω in the game $\mathbb{G}(\mathscr{A}, G)$ is defined as follows: for atomic positions $(\overset{\exists}{\forall}{}^{\pm 1} q, v)$ we set $\Omega(\overset{\exists}{\forall}{}^{\pm 1} q, v)$ is 1 iff $q \in F$, otherwise it is 0. For the remaining positions we set 3 so that their color is irrelevant. The starting game position v_0' in $\mathbb{G}(\mathscr{A}, G)$ is $(\delta(q_0, e(v_0)), v_0)$.

Definition 2.4 (*Acceptance by the automaton*). We say the automaton $\mathscr{A}$ *accepts* a graph G under the strategy s if s is a winning strategy for player 0 in the game $\mathbb{G}(\mathscr{A}, G)$. Such a strategy s is called *accepting*. We say $\mathscr{A}$ accepts graph G if there exists a strategy s such that $\mathscr{A}$ accepts G under s.

Note: If we have a formula ψ that is an immediate subformula of two distinct formulas ϕ and ϕ', in the game graph for each vertex v of G there will be an edge to (ψ, v) from both (ϕ, v) and (ϕ', v). One might think this unnatural, since going from (ϕ, v) to (ψ, v) is something different than going from (ϕ', v) to (ψ, v). However, in the general case we are dealing with strategies with memory and such a strategy can remember where it was before coming to (ψ, v). By using the memoryless determinacy theorem it turns out that this information is not necessary.

It can be shown that one-way alternating automata on graphs have a certain finite graph property, that is, if a given one-way alternating automaton accepts any kind of graph, it also accepts a finite graph. This, however, is not the case when speaking of two-way alternating automata. We will conclude this section with an example of an automaton that accepts only infinite graphs.

As an example, consider the following two-way automaton:

$$\mathscr{A} = \langle \{q_0, q_1\}, q_0, \{a\}, \delta, \{q_0\} \rangle$$

$$\delta(q_0, a) = \exists^{+1} q_0 \wedge \forall^{-1} q_1 \quad \delta(q_1, a) = \forall^{-1} q_1$$

Let us examine the game $\mathbb{G}(\mathscr{A}, G)$, where $G = \langle \mathbb{N}, \{(n, n+1)\colon\ n \in \mathbb{N}\}, e, 0 \rangle$, such that $e(n) = a$ for all $n \in \mathbb{N}$. Consider first the following example play. The play starts in formula $\exists^{+1} q_0 \wedge \forall^{-1} q_1$ at vertex 0. This is a position for player 1, let's assume she chooses the subformula $\exists^{+1} q_0$. Now player 0 has to choose a neighboring (in G) vertex along a forward edge. She has to choose 1; the position is now $\exists^{+1} q_0 \wedge \forall^{-1} q_1$ at vertex 0. This goes on until, say, we reach 10. Let's assume that this time player 1 chooses the subformula $\forall^{-1} q_1$. Now it is her choice to choose a neighboring vertex in G, along a backward edge; she has to choose vertex 9—there is no other backward edge from 10. The play then goes on through positions $(\forall^{-1} q_1, 9), \ldots, (\forall^{-1} q_1, 0)$ in which last position player 1 loses for a lack of possible moves.

In the game $\mathbb{G}(\mathscr{A}, G)$ there are essentially two kinds of play: a finite play like the one above, where player 0 wins, or an infinite one where player 1 always chooses the subformula $\exists^{+1} q_0$. The play goes through positions $(\exists^{+1} q_0 \wedge \forall^{-1} q_1, 0), (\exists^{+1} q_0, 0), \ldots, (\exists^{+1} q_0 \wedge \forall^{-1} q_1, k), (\exists^{+1} q_0, k), \ldots$. Since q_0 is an accepting state, the only color appearing infinitely often in this play is 0, thus, again, player 0 wins.

So we see that in the game $\mathbb{G}(\mathscr{A}, G)$ player 0 has a winning strategy, in other words, $\mathscr{A}$ accepts G.

Fact 2.1. For any graph G, the automaton $\mathscr{A}$ does not accept in a vertex v_1 and state q_0 if

(1) v_1 is contained in a sequence $v_1 v_2 \ldots$ where for all $i \in \{1, 2, \ldots\}$, (v_{i+1}, v_i) is an edge in G.

(2) v_1 is not contained in a sequence $v_1 v_2 \ldots$ where for all $i \in \{1, 2, \ldots\}$, (v_i, v_{i+1}) is an edge in G and $\mathscr{A}$ accepts in v_i and q_0.

Corollary 2.1. *$\mathscr{A}$ accepts only infinite graphs.*

Proof. Assume $\mathscr{A}$ accepts a finite graph starting in vertex v_1. By item 2 in Fact 2.1, there must exist an appropriate infinite sequence, which in a finite graph can only be cycle. This, however would contradict item 1 in Fact 2.1. □

2.3. Automaton paths

Let us fix a two-way alternating automaton $\mathscr{A}$. Given a play r in the game $\mathbb{G}(\mathscr{A}, G)$ we define $\tilde{r}$ as the sequence of state-vertex pairs visited in the play r. For instance, in the example above, for the first play r, we have $\tilde{r} = (q_0, 0)(q_1, 1) \ldots (q_0, 10)(q_1, 9)(q_1, 8) \ldots (q_1, 0)$. The following is a key definition:

Definition 2.5 (*Automaton path*). Let r be a play consistent with the strategy s in $\mathbb{G}(\mathscr{A}, G)$. Any contiguous subsequence of $\tilde{r}$ is called an *automaton path* in G, s.

We use $\mathrm{AP}(G, s)$ to denote the set of all automaton paths in G, s. Sometimes we shall omit the word automaton and simply say path, where confusion can arise we shall distinguish automaton paths from graph paths. The length of a path p is denoted by $|p|$. We say the path p leads from (q, v) to (q', v'), denoted as $(q, v) \to^p (q', v')$ if $(q, v)p(q', v')$ is an automaton path in the game obtained from $\mathbb{G}(\mathscr{A}, G)$ by setting the starting game position to $(\delta(q), v)$. Note that this does not mean that $p \in \mathrm{AP}(G, s)$, for this we need also (q, v) to be *accessible*, that is, $(q_0, v_0) \to (q, v)$ must hold.

By $p(i)$ we denote the ith element of the path p, that is (q_i, v_i). In particular, $p = p(1)p(2) \ldots p(|p|)$. A path p is a *sub-path* of p', written as $p \sqsubseteq p'$, if p is a contiguous subsequence of p'. We define $\|p\|_Q$ as the set of states visited in p and $\|p\|_V$ as the set of vertices visited in p. Consider the function num assigning 0 to accepting states and 1 to the remaining states. For $R \subseteq \{0, 1\}$, we write $(q, v) \to_R (q', v')$ if $(q, v) \to^p (q', v')$ for some p such that $\mathrm{num}(\|p\|_Q) = R$.

We say that a finite path p *ends well* under the strategy s if it corresponds to a finite play r in $\mathbb{G}(\mathscr{A}, G)$ which is winning for 0, that is one where player 1 cannot make a move. We will now rephrase the acceptance condition in terms of automaton paths: the automaton $\mathscr{A}$ accepts a graph G under the strategy s iff every maximal (in terms of $\sqsubseteq$) finite path ends well under s and every infinite path $(q_1, v_1), (q_2, v_2), \ldots$ visits the accepting states infinitely often.

Corollary 2.2. *If the automaton $\mathscr{A}$ accepts the graph G under s there is no cycle where only odd states appear.*

2.4. Tree unraveling

A very important concept that will be used here is the tree unraveling of a graph. By a *two-way path* in a graph $G = \langle V, E, \Sigma, e, v_0 \rangle$ we mean any sequence of neighboring vertices, that is, any sequence $v_1 \cdot \ldots \cdot v_i$ such that $(v_j, v_{j+1}) \in E$ or $(v_{j+1}, v_j) \in E$ for $j \in \{1, \ldots, i-1\}$.

Definition 2.6 (*Tree unraveling*). Given a graph $G = \langle V, E, \Sigma, e, v_0 \rangle$, its *tree unraveling* is the graph $\mathrm{Un}(G) = \langle V', E', \Sigma, e', v_0 \rangle$, where the set of vertices V' is the set of finite two-way paths in G starting in v_0, the set of edges is defined $E' = \{((\pi \cdot v), (\pi \cdot v \cdot w))^i \colon (\pi \cdot v \cdot w) \in V', (v,w)^i \in E\}$ and the labeling is set as $e'(\pi \cdot v) = e(v)$.

The *depth* of a vertex in the unraveling is its distance from the root of the tree (across edges in both directions), or, in other words, the length of the path denoted by this vertex. For two vertices π_1 and π_2 in $\mathrm{Un}(G)$ (that is, two-way paths in G), we say π_1 is a successor of π_2 if π_2 is an initial segment of the path π_1. Note that this is a *two-way* tree, that is, edges between a son and father can be either forward or backward. For a tree T we use $T|_\pi$ to signify the subtree of T rooted at π, $T|^i$ the fragment of T up to depth i.

Having a tree unraveling we define the canonical projection $\Pi : V' \to V$ which maps a tree vertex πv onto v. We define the *game projection* $\Pi^{\mathbb{G}}$ from game positions in $\mathbb{G}(\mathscr{A}, \mathrm{Un}(G))$ onto game positions in $\mathbb{G}(\mathscr{A}, G)$ so that $\Pi^{\mathbb{G}}(x, (\pi, v)) = (x, v)$.

Definition 2.7 (*Strategy unraveling*). We say the strategy $\mathrm{Un}(s)$ is the *unraveling of strategy* s if $\Pi \circ \mathrm{Un}(s) = s \circ \Pi^{\mathbb{G}}$.

Lemma 2.1. *An automaton path p is in* $\mathrm{AP}(G,s)$ *iff there exists a path* $p' \in \mathrm{AP}(\mathrm{Un}(G), \mathrm{Un}^{\mathbb{G}}(s))$ *such that* $\Pi(p') = p$.

Proof. It suffices to prove the lemma for paths starting in q_0, v_0. The proof is by induction on the length of the path. □

Corollary 2.3. *The automaton $\mathscr{A}$ accepts a graph G under strategy s iff $\mathscr{A}$ accepts* $\mathrm{Un}(G)$ *under* $\mathrm{Un}(s)$.

3. The finite graph problem

Corollary 2.1 is a motivation for the following problem: "does a given alternating two-way automaton accept some finite graph?" Let us denote this problem by *FIN-ALT*.

We shall now define the concept of an automaton *signature*, used in the key Theorem 3.1 of this paper. Consider a graph $G = \langle V, E, \Sigma, e, v_0 \rangle$ accepted by $\mathscr{A}$.

Definition 3.1 (*Signature*). Let $q \in Q$, $v \in G$, $p \in \mathrm{AP}(G,s)$ and $i \in \mathbb{N}$.

- $\mathrm{Sig}(p) \in \mathbb{N} \cup \{\infty\}$ is $max\{j\colon \; \|p[1..j]\|_Q \cap F = \emptyset\}$
- $\mathrm{Sig}^{G,s}(q,v) \in \mathbb{N} \cup \{\infty\}$ is $max\{\mathrm{Sig}(p)\colon \; p \in \mathrm{AP}(G,s),\, p_1 = (q,v)\}$.
- $\mathrm{Sig}^{G,s}(v) \in (\mathbb{N} \cup \{\infty\})^Q$ is $(\mathrm{Sig}^{G,s}(q,v))_{q \in Q}$

If G and s are clear we shall simply write $\mathrm{Sig}(q,v)$ instead of $\mathrm{Sig}^{G,s}(q,v)$. Intuitively, $\mathrm{Sig}(q,v)$ gives the longest possible length of an automaton path consisting of non-accepting states starting in (q,v). We shall assume $\mathrm{Sig}(q,v) = \infty$ if there is no such bound. We say $\mathscr{A}$ accepts G with a signature *bounded* by $N \in \mathbb{N}$, if there exists a strategy s such that for each state q of $\mathscr{A}$ and each vertex v of G, $\mathrm{Sig}^{G,s}(q,v) \leqslant N$.

For $k \in \mathbb{N}$, we $\mathscr{A}$ accepts a graph G with a *signature bounded by* k, if there is some strategy s such that $\mathrm{Sig}_i(q,v) \leqslant k$ for all q, v and odd i. The automaton accepts a graph with *bounded signature* iff for some k it accepts the graph with a signature bounded by k.

By $|\mathscr{A}|$ we denote the *size* of the automaton $\mathscr{A}$, that is either one of: $\log(|\Sigma|)$, the number of states in $\mathscr{A}$ or the biggest size of a formula in the transition function of $\mathscr{A}$, whichever is greater. The following theorem is our main technical result:

Theorem 3.1 (Small signature theorem). *For any alternating two-way automaton* $\mathscr{A} = \langle Q, q_0, \Sigma, \delta, F\rangle$ *the following three conditions are equivalent*:
(1) *$\mathscr{A}$ accepts some finite graph*
(2) *$\mathscr{A}$ accepts some two-way tree with a bounded signature*
(3) *$\mathscr{A}$ accepts some two-way tree of degree $4|Q|$ with a signature bounded by a constant doubly exponential in $|\mathscr{A}|$*

Let us fix the automaton $\mathscr{A} = \langle Q, q_0, \Sigma, \delta, F\rangle$. The proof of this theorem is long and will be distributed across three subsection.

3.1. *Proof of* $1 \Rightarrow 2$

The two-way tree in question will be the unraveling of a graph accepted by $\mathscr{A}$. First we shall state two lemmas.

Lemma 3.1. *If the automaton $\mathscr{A}$ accepts the finite graph $G = \langle V, E, \Sigma, e, v_0\rangle$ under s, then for every $q \in Q$, $v \in V$ we have* $\mathrm{Sig}^{G,s}(q,v) \leqslant |V||Q|$.

Proof. Assuming the contrary, we would obtain a cycle of non-accepting states. □

Lemma 3.2. *The tree unwinding does not increase the signature, i.e.*

$$\mathrm{Sig}^{\mathrm{Un}(G),\mathrm{Un}(s)}(q,(\pi v)) \leqslant \mathrm{Sig}^{G,s}(q,v)$$

Proof. This follows from Lemma 2.1. □

For the proof of $1 \Rightarrow 2$, assume $\mathscr{A}$ accepts the graph G. Then $\mathscr{A}$ accepts $\mathrm{Un}(G)$ under the strategy $\mathrm{Un}(s)$ (Lemma 2.3). Moreover, for every vertex v of the tree $\mathrm{Un}(G)$ and every state $q \in Q$ we have $\mathrm{Sig}^{\mathrm{Un}(G),\mathrm{Un}(s)}(q,v) \leqslant \mathrm{Sig}^{G,s}(q,\Pi(v)) \leqslant |V||Q|$. The first inequality is due to Lemma 3.2, the second due to 3.1.

3.2. *Proof of* $3 \Rightarrow 1$

We will, in fact, prove the stronger implication $2 \Rightarrow 1$. Take a tree T accepted by $\mathscr{A}$ under the strategy s. First we will cut out some branches. Take a vertex of v of T

and out of the set of its successors S_v, consider the set S'_v of those vertices $w \in S_v$ such that $s(v, \exists^{\pm 1} q) = (w, \phi)$ for some positions $(v, \exists^{\pm 1} q), (w, \phi)$ in the game $\mathbb{G}(\mathscr{A}, G)$. The size of S'_v is at most $4|Q|$. It is now an easy observation that we can, without violating acceptance, remove from T all the subtrees of v not rooted in S'_v. This process leads us to a new tree of degree at most $4|Q|$.

Having such a tree we can think of a bounded representation of strategies. A strategy in a given vertex consists of: (a) choosing subformulas for compound formulas and (b) choosing neighboring vertices for existential atoms. Obviously, there are only exponentially many (with respect to $|\mathscr{A}|$) different strategies possible in any given vertex. Call this number of strategies $\tilde{\mathscr{A}}$.

With every vertex v of the tree $T = \langle V, E, \Sigma, e, \lambda \rangle$ we shall associate two pieces of information constituting the *type* of v: the strategy s in the vertex v, $e(v)$ and $\mathrm{Sig}(v)$. Because strategies are encoded by a number from 1 to $\tilde{\mathscr{A}}$ and since by assumption the signature is bounded by some N, there exists a finite number of vertex types. We can thus find such a number $i \leqslant N \cdot \tilde{\mathscr{A}}$ that all vertex types in the subtree $T|^{i+1}$ appear already in the subtree $T|^i$.

Let $f : T|^{i+1} \to T|^i$ be any function such that f restricted to $T|^i$ is the identity mapping and for every vertex $v \in T|^{i+1}$, v and $f(v)$ have the same type. Such a function exists by assumption on i. Consider now the following graph $T' = \langle T|^i, E', \Sigma, e \circ f, \lambda \rangle$ resulting from "looping" the tree T on the level i. We define the set of edges E' of the graph T' as follows: $E' = \{(f(v), f(v')) \colon (v, v') \in E\}$, where E is the set of edges of the original tree T.

Definition 3.2 (*Pseudo-signature*). A function $\Sigma : Q \times V \to \{0, \ldots, N\}$ is called a *pseudo-signature* for G, s if for every $(q, v)(q', v') \in \mathrm{AP}(G, s)$, Σ satisfies $\Sigma(q', v') \leqslant \Sigma(q, v)$ and the inequality is proper if $q \notin F$.

Lemma 3.3. *$\mathscr{A}$ accepts a finite graph G under the strategy s iff there exists a pseudo-signature for G, s.*

Proof. For the left to right implication it is sufficient to notice that the signature Sig is a pseudo-signature. For the other direction, one has to show that a pseudo-signature majorizes Sig, so that each state with odd priority appears at most some fixed number of times before an even priority state appears. □

It is an easy exercise to show that the function $\mathrm{Sig}^{T,s}$ is a pseudo-signature for the graph T', thus proving that $\mathscr{A}$ accepts the finite graph T'.

Using the above technique and the assumption 3, we obtain:

Corollary 3.1 ("Small" model theorem). *If an automaton accepts some finite graph, then it accepts a graph of size triply exponential in $|\mathscr{A}|$.*

3.3. Proof of $2 \Rightarrow 3$

Let T, s be as in condition 2 of Theorem 3.1. By assumption we know there exists a certain, if perhaps difficult to estimate, bound on the signature. We will now modify

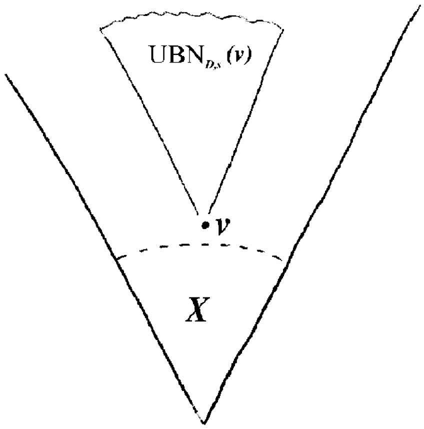

Fig. 1. The tree D.

the tree T in such a way as to make this constant tractable. We use the same technique as in the previous subsection to ensure that T is of degree at most $4|Q|$.

First, we shall introduce the following definitions. We say a path $p \in \mathrm{AP}(G,s)$ is *bad* iff $\|p\|_Q \cap F = \emptyset$. We write $v \rightarrow_{\{1\}} w$ if there exist states $q, q' \in Q$ such that $(q', v) \rightarrow_{\{1\}} (q, w)$. We define the *upper bad neighborhood* $\mathrm{UBN}_{D,s}(v)$ of a vertex v in the tree D under the strategy s as the set of v's "bad-accessible" (both ways) successors, i.e. $\mathrm{UBN}_{D,s}(v) \equiv_{def} \{w \in D|_v\colon\ w \rightarrow_{\{1\}} v \vee v \rightarrow_{\{1\}} w \text{ in } D, s\}$

Let M be a constant whose exact size depending on the size of the automaton $\mathscr{A}$ we will estimate later in this paper. For a tree D, strategy s and vertex v of the tree D, denote the following property as $(*)$:

$$(*)\mathrm{UBN}_{D,s}(v) \subseteq D|_v^M$$

We are now going to construct a sequence of trees and accepting strategies (D^0, s^0), $(D^1, s^1), \ldots$ such that

(1) Each two tree-strategy pairs (D^i, s^i) and (D^j, s^j) are identical up to depth $min(i,j) - 1$

(2) Property $(*)$ holds for vertices v in (D^i, s^i) of depth less than i.

We will define the trees inductively with respect to i. Let (D^0, s^0) be simply (T, s). The above conditions obviously hold for (D^0, s^0) since there are no vertices of depth less than zero. Assume now that we have constructed (D^i, s^i). We will define (D^{i+1}, s^{i+1}) by iterating the following Lemma 3.4 for successive vertices of depth i. Of course the conditions in the lemma are satisfied by (D^0, s^0).

Definition 3.3. For any downwards closed set X of vertices from D, we say D, s is *X-OK* iff:

(1) $\mathscr{A}$ accepts D under s

(2) $\mathrm{UBN}_{D,s}(w)$ is finite for every vertex w of the tree D.
(3) $(*)$ holds for vertices in X.

Lemma 3.4. *Let X be a downwards closed set of vertices in D. If D, s is X-OK then for any successor v of any vertex in X, there exists a tree D_v and strategy s_v identical with D, s on the set $X \cup \{v\}$ such that D_v, s_v is $X \cup \{v\}$-OK.*

Proof. It is enough to show that if $(*)$ does not hold for v, we can find a tree D' and strategy s', identical with D, s on the set $X \cup \{v\}$ such that D', s' is X-OK and moreover the following inequality holds (the cardinality of both sets is finite by assumption 3 of Definition 3.3):

$$|\mathrm{UBN}_{D',s'}(v)| < |\mathrm{UBN}_{D,s}(v)|$$

Iterating this process we arrive at D_v, s_v from the conclusion of the lemma.

We will consider a certain equivalence relation $\simeq$ defined on $\mathrm{UBN}_{D,s}(v)$. Vertices equivalent under this relation are in a sense interchangeable (along with their subtrees). First, we introduce a symbol $O(v,w)$ which, for any two vertices v, w in the tree D, denotes the set $\{(q,R,q') \in Q \times P(\{0,1\}) \times Q\colon\ \exists p \in \mathrm{AP}(D,s).(q,v) \rightarrow^{p}_{R} (q',w)\}$. Using this notation, we write $w \simeq w'$ iff all the following conditions hold:
(1) $s(w) = s(w')$
(2) $O(w,w) = O(w',w')$
(3) $O(v,w) = O(v,w')$
(4) $O(w,v) = O(w',v)$
(5) $e(w) = e(v)$
Now is the time to calculate M: it is the number of abstraction classes of the relation $\simeq$, that is, at most $|\Sigma| \cdot \tilde{\mathscr{A}} \cdot (2^{4|Q|^2})^3$. In other words, M is exponential with respect to $|\mathscr{A}|$. Assume now that the depth of $\mathrm{UBN}_{D,s}(v)$ is greater than M. In such a case we can find two vertices $w, w' \in \mathrm{UBN}_{D,s}(v)$, $w < w'$ such that $w \simeq w'$. Now take for D' the tree resulting from substituting $D|_w$ for $D|_{w'}$, and let s' be the strategy s restricted to the new, smaller tree.

Claim 3.1. *D', s' is X-OK and $|\mathrm{UBN}_{D',s'}(v)| < |\mathrm{UBN}_{D,s}(v)|$*

We need two lemmas to prove this claim.

Definition 3.4 (*Clean path*). We say a path $p \in \mathrm{AP}(D',s')$ is *clean* iff it does not visit w'. Clean paths can be either *upper*, that is contained in $D'|_{w'}$, or *lower*—the remainder.

Lemma 3.5. *For any state $q \in Q$ and $R \subseteq \{0,1\}$, if $(q_1,v) \rightarrow_R (q_2,w')$ in D', s', then in D, s both $(q_1,v) \rightarrow_R (q_2,w)$ and $(q_1,v) \rightarrow_R (q_2,w')$. This is also true if we replace $\rightarrow$ with $\leftarrow$.*

Proof. We only prove this for $\rightarrow$. Let p be such that $(q_1,v) \rightarrow^{p} (q_2,w')$ in D', s'. The proof is by induction on the number of times w' appears in p.

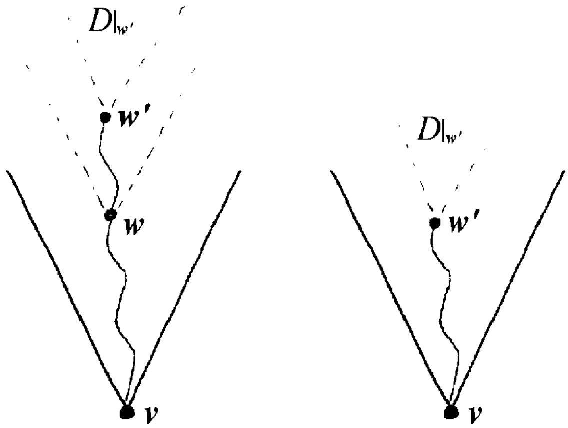

Fig. 2. Before and after the cut ($D|_v$ and $D'|_v$).

If p is clean then the conclusion is obvious, using condition 3 of $w \simeq w'$. Otherwise $p = p_1(q', w')p_2$, where p_2 is clean. Let us just consider the case where p_2 is upper, the proof of the other is analogous. If p_2 is upper then $(q', w') \rightarrow^{p_2} (q_2, w')$ in D, s. By condition 2 of $w \simeq w'$ we have $(q', w) \rightarrow^{p_3} (q_2, w)$ in D, s for some path p_3 such that $\mathrm{num}(\|p_3\|_Q) = \mathrm{num}(\|p_2\|_Q)$. By induction hypothesis, for some paths p_A and p_B such that $\mathrm{num}(\|p_A\|_Q) = \mathrm{num}(\|p_B\|_Q) = \mathrm{num}(\|p_1\|_Q)$, we have both $(q_1, v) \rightarrow^{p_A} (q', w')$ and $(q_1, v) \rightarrow^{p_B} (q', w)$ in D, s. The paths $p_A(q', w')p_2$ and $p_B(q', w)p_3$ give us the desired assertion. $\square$

Lemma 3.6. *For any u not greater than v, if $(q, u) \rightarrow_R (q', u')$ in D', s' then $(q, u) \rightarrow_R (q', u')$ in D, s.*

Proof. Take a path p such that $(q, u) \rightarrow^p (q', u')$. If p is clean then we are done. Otherwise, consider the first occurrence of v and the last occurrence of w' in p: $p = p_1(q_1, v)$ $p_2(q_2, w')p_3$. If u is below w' then by Lemma 3.5 there is a path p'_2 visiting the same colors as p_2 such that $(q_1, v) \rightarrow^{p'_2} (q_2, w)$ in D, s. We obtain the desired result by replacing p_2 with p'_2, since the remaining paths are good in D, s. If u is above w', we use w' instead of w. $\square$

Corollary 3.2. *Any pair (q, u) reachable in D', s' is reachable in D, s. Thus for clean paths p, if $p \in \mathrm{AP}(D', s')$ then $p \in \mathrm{AP}(D, s)$.*

Proof. We use Lemma 3.6 with (q_0, v_0) as (q, u). For the second part, if p is clean and $p(1)$ is reachable (by the first part), then $p \in \mathrm{AP}(D, s)$. $\square$

To finish the proof of Lemma 3.4, we prove Claim 3.1

- To show that $\mathscr{A}$ accepts the tree D' under s' we need to prove there is no infinite bad path p in $\mathrm{AP}(D', s')$ and no path therein ends badly. The second part follows from

Corollary 3.2. For the first part, take an infinite bad path. If it were to visit w' only finitely often, then its infinite bad suffix would, by Corollary 3.2, be in $\mathrm{AP}(D,s)$, which is a contradiction. Then p must be of the form $(q_1,v_1)p_1(q_2,w')\dots p_n(q_n,w')$, where all paths p_i are clean and $q_n=q_i$ for some $i<n$. By replacing each upper bad p_i with an p_i' starting and ending in w (obtained from the fact that $w\simeq w'$), we would obtain a bad cycle reachable in D,s.

- Secondly, we want to show that $\mathrm{UBN}_{D',s'}(u)$ is finite for each vertex u in the tree D',s'. Take any vertex u' visited by some bad path p going through u. If the part of p between u and u' is clean then we are done by Corollary 3.2. Otherwise u' is in $\mathrm{UBN}_{D',s'}(w')$. All we need to do is show that u' is in $\mathrm{UBN}_{D,s}(w')\cup\mathrm{UBN}_{D,s}(w)$, since these sets are bounded by assumption of D,s being X-OK.

  Assume that $(q,u)\rightarrow^{p}_{\{1\}}(q',w')$ (the case where $(q',w')\rightarrow_{\{1\}}(q,u)$ is symmetric). Without lessening of generality, we can assume p is clean. If it is lower, we replace w' by w in p, otherwise we leave it as is. It follows from Corollary 3.2 that p is in $\mathrm{AP}(D,s)$. This means $u\in\mathrm{UBN}_{D,s}(w')\cup\mathrm{UBN}_{D,s}(w)$.
- It follows from Lemma 3.6, that for $u\in X\cup\{v\}$ we have $\mathrm{UBN}_{D',s'}(u)\subseteq\mathrm{UBN}_{D,s}(u)$, in particular, the cardinality of the first set for $u=v$ is smaller than the cardinality of the second (since we cut out vertices from $\mathrm{UBN}_{D,s}(v)$). □

Having thus proven 3.4 we can conclude by using the trees $(D^0,s^0),(D^1,s^1),\dots$ to prove the $2\Rightarrow 3$ implication. Since the trees $(D^i,s^i),(D^j,s^j)$ are identical up to depth $min(i,j)-1$, we can define the limit tree D and strategy s which are identical with each D^i,s^i up to depth $1-i$. Now take a vertex $v\in D$ and let $n=|v|+M$. Since $\mathrm{UBN}_{D^n,s^n}(v)$ contains only vertices of depth less than n, we see that $\mathrm{UBN}_{D,s}(v)\subseteq D|_v^M$, or, in other words, $(*)$ holds for all vertices of D,s. It is now a trivial observation that the maximal length of a bad path contained within $D|_v^M$ is at most exponential with respect to M, otherwise we would have a cycle. Thus the length of all bad paths in D is at most doubly exponential in $|\mathscr{A}|$ and consequently so is the signature.

3.4. The FIN-ALT problem is decidable

Armed with Theorem 3.1 we are ready to show the decidability of the *FIN-ALT* problem.

Theorem 3.2. *The FIN-ALT problem is decidable in 2EXPTIME.*

Proof. Recall from Theorem 3.1 that $\mathscr{A}$ accepts some finite graph iff it accepts a tree of degree $4|Q|$ with the signature bounded by M, which is doubly exponential on $|\mathscr{A}|$ (a tree satisfying condition 3 of the theorem).

Given an automaton $\mathscr{A}$ we shall construct a nondeterministic automaton $\mathscr{A}'$ on trees over the alphabet Σ which accepts precisely those trees that satisfy condition 3. The automaton $\mathscr{A}'$ guesses a strategy and a function $\Sigma: V\times Q\rightarrow\{1,\dots,M\}$ and then checks if Σ is a pseudo-signature. Having guessed it, the automaton moves down the tree, remembering the signature and strategy of the parent vertex to check whether the conditions of Definition 3.2 are satisfied. To remember the signature and strategy, the automaton needs a doubly exponential number of states.

Thus the emptiness problem for $\mathscr{A}'$ (in terms of nondeterministic automata) is equivalent to the emptiness of $\mathscr{A}$ (in terms of two-way alternating automata). Since the automaton $\mathscr{A}'$ checks only a local consistency, it has the following nice property: every run of $\mathscr{A}'$ is accepting. It can be proved that for such automata, indeed even for Büchi nondeterministic automata, the emptiness problem is decidable in polynomial time [12] and thus we obtain the time in the theorem's conclusion. □

4. The μ-calculus with backward modalities

In this section we introduce the modal μ-calculus with backward modalities. First, however, we will sketch out a new form of our two-way automaton which we will be convenient in the μ-calculus reduction.

4.1. Enhanced automata

For a briefer notation we will add two new mechanisms to two-way alternating automata which do not expand their expressive power. Let $\circ Q$ be the set $\{\circ q\colon\ q\in Q\}$. Our new *enhanced automata* are identical to alternating two-way automata, save they have a more complicated transition function. In an enhanced automaton, δ assigns to each state-label pair $(q,a)\in Q\times\Sigma$ a formula $\delta(q,a)$ built from atoms of the form $\exists Q$, $\forall Q$ (as before) and, additionally, $\circ Q$ and *true* and *false*.

We interpret *true* and *false* as follows: when player 0 reaches *true*, he wins, while when he reaches *false*, he looses; the reverse holds for player 1. On the other hand, $\circ q$ means the automaton stays in the same vertex, only changes its state to q.

Lemma 4.1. *The finite graph problem for enhanced automata with the Büchi condition is decidable in double exponential time with respect to the size of the automaton.*

Proof. The proof is essentially the same as for alternating two-way automata. The only difference is in Theorem 3.1. The constant M is still doubly exponential, it is however multiplied by $|Q|$—this is how long an enhanced automaton can stay in one vertex in bad states under a memoryless winning strategy. □

4.2. The μ-calculus

Let $AP=\{p,q,\dots\}$ be a set of atomic propositions, and let $VAR=\{X,Y,\dots\}$ be a set of propositional variables.

Definition 4.1 (*Formulas of the calculus*). The set of formulas of the μ-calculus with backward modalities is the smallest set such that:
- Every atomic proposition $p\in AT$ and its negation $\neg p$ are formulas
- Every variable $X\in VAR$ is a formula

- If α and β are formulas and $X \in VAR$ then the following are formulas:

$$\alpha \vee \beta, \alpha \wedge \beta, \exists^{+1}\alpha, \forall^{+1}\alpha, \exists^{-1}\alpha, \forall^{-1}\alpha, \mu X.\alpha, \nu X.\alpha$$

We call μ and ν, respectively, the least and greatest fix-point operators. We will write ϑ to signify any one of the two operators. Formulas of the calculus are interpreted in so-called *Kripke structures*.

Definition 4.2 (*Kripke structure*). A *Kripke structure* $K = \langle V, E, S \rangle$ consists of a graph $\langle V, E \rangle$ along with a function $S : V \to P(AP)$ which assigns to each vertex the set of atomic propositions true in that vertex.

Let $K = \langle V, E, S \rangle$ be a Kripke structure, v a *valuation*, i.e. any function $v : VAR \to P(V)$. As usual, we define $v[W/X]$ as the valuation obtained from v by substituting the set $W \subseteq V$ for the variable X. The interpretation of a formula ϕ in a given Kripke structure under the valuation v, written as $\phi^K[v]$, is defined inductively as follows:
- For atomic propositions $p \in AP$, $p^K[v] = \{u \in V : p \in S(u)\}$
- For variables $X \in VAR$, $X^K[v] = v(X)$
- $(\phi_1 \wedge \phi_2)^K[v] = \phi_1^K[v] \cap \phi_2^K[v]$
- $(\phi_1 \vee \phi_2)^K[v] = \phi_1^K[v] \cup \phi_2^K[v]$
- $(\exists^k \phi)^K[v] = \{u \in V : \exists w \in V.(u,w)^k \in E \wedge w \in \phi^K[v]\}, \ k \in \{1, -1\}$
- $(\forall^k \phi)^K[v] = \{u \in V : \forall w \in V.(u,w)^k \in E \Rightarrow w \in \phi^K[v]\}, \ k \in \{1, -1\}$
- $(\mu X.\phi)^K[v] = \bigcap \{V' \subseteq V : \phi^K[v[V'/X]] \subseteq V'\}$
- $(\nu X.\phi)^K[v] = \bigcup \{V' \subseteq V : V' \subseteq \phi^K[v[V'/X]]\}$

4.3. Automata on models

In this section we sketch the correspondence between the μ-calculus and enhanced automata. Let $AP(\phi)$ be the set of atomic predicates $p \in AP$ occurring in ϕ. Let $\Sigma_\phi = P(AP(\phi))$.

Definition 4.3 (*Encoding*). The *encoding* of a Kripke structure $K = \langle V, E, S \rangle$ from vertex $v_0 \in V$ is the graph $G(K, v_0) = \langle V, E, \Sigma_\phi, e, v_0 \rangle$ where $e : V \to \Sigma_\phi$ is the restriction of S to Σ_ϕ.

Let ϕ be a sentence of the μ-calculus. We will construct an enhanced automaton $\mathscr{A}_\phi$ on graphs that will recognize the encodings of models for ϕ. By $cl(\phi)$ we mean the smallest set of formulas closed under subformulas such that $\phi \in cl(\phi)$ and if $\vartheta X.\varphi(X) \in cl(\phi)$ then $\varphi(\vartheta X.\varphi(X) \in cl(\phi))$. Let $\mathscr{A}_\phi = \langle cl(\phi), \phi, \Sigma_\phi, \delta, F \rangle$. The transition function δ is defined as follows:
- $\delta(p, \Sigma) = true$ if $p \in \Sigma$, *false* otherwise.
- $\delta(\neg p, \Sigma) = false$ if $p \in \Sigma$, *true* otherwise.
- $\delta(\phi_1 \vee \phi_2, \Sigma) = \circ \phi_1 \vee \circ \phi_2$
- $\delta(\phi_1 \wedge \phi_2, \Sigma) = \circ \phi_1 \wedge \circ \phi_2$
- $\delta(\vartheta X.\varphi(X), \Sigma) = \circ \varphi(\vartheta X.\varphi(X))$

- $\delta(\exists^k \varphi, \Sigma) = \exists^k \varphi$ for $k \in \{1, -1\}$
- $\delta(\forall^k \varphi, \Sigma) = \forall^k \varphi$ for $k \in \{1, -1\}$

We fix the coloring Ω so as to satisfy the following conditions:

- If the variable Y occurs freely in $\vartheta X.\phi_1(X)$ then $\Omega(\vartheta X.\phi_1(X)) > \Omega(\vartheta Y.\phi_2(Y))$.
- $\Omega(\vartheta X.\varphi(X))$ is even if and only if $\vartheta = \nu$
- Formulas not beginning with ϑ have a color no smaller than any fix-point formula.

Lemma 4.2 (Vardi [14]). *For every sentence ϕ and every Kripke structure $K, v_0 \in \phi^K$ if and only if $\mathscr{A}_\phi$ accepts the encoding $G(K, v_0)$.*

Corollary 4.1 (of Lemma 4.2). *A sentence ϕ of the μ-calculus has a finite model if and only if the automaton $\mathscr{A}_\phi$ accepts some finite graph.*

We say a sentence of the μ-calculus is $\nu\mu$, if it has no sub-formulas $\phi_\nu = \nu X.\phi_1(X)$ and $\mu Y.\phi_2(Y)$ such that Y occurs freely in ϕ_ν. Now it is easy to see that in Lemma 4.2, $\nu\mu$ sentences are translated into enhanced automata whose acceptance condition is equivalent to the Büchi condition (the coloring consists of first a number of small even priorities, then larger odd priorities).

Theorem 4.1. *The problem whether a $\nu\mu$ sentence of the μ-calculus with backward modalities has a finite model is in 2EXPTIME.*

Proof. Given a $\nu\mu$ sentence ϕ we construct an equivalent enhanced automaton $\mathscr{A}_\phi$ and solve this instance of the enhanced *FIN-ALT* problem. All three components of $|\mathscr{A}_\phi|$ are polynomial with respect to the length of the formula ϕ. $\square$

5. Closing remarks

The main result of this paper is a proof of the decidability of the finite graph problem for two-way alternating automata with the Büchi condition. This can be used to prove the decidability of the finite model problem for a certain sub-logic of the propositional μ-calculus with backward modalities.

The proof is based on Theorem 3.1, which uses the concept of signature. In this theorem, implications $1 \Rightarrow 2$ and $3 \Rightarrow 1$ can be easily generalized for automata with an arbitrary parity condition. However, it remains an open problem whether such a generalization is possible for the implication $2 \Rightarrow 3$.

It seems that a decidability proof for the whole problem is desirable for ends other than the finite model problem of the μ-calculus with backward modalities. Two-way alternating automata are used in paper [3] to decide the satisfiability of formulas of the so-called Guarded Fragment with fixed points. It can be supposed that the finite graph problem for the full parity condition can be applied to solving the open problem of whether the finite model property for formulas of the Guarded Fragment with fixed points is decidable.

References

[1] J. van Benthem, H. Andreka, I. Nemeti, Modal logics and bounded fragments of predicate logic, J. Philos. Logic 27 (1998) 217–274.

[2] E.A. Emerson, C. Jutla, Tree automata, mu-calculus and determinacy, Proc. 32th IEEE Symp. on Foundations of Computer Science, 1991, pp. 368–377.

[3] E. Grädel, On the restraining power of guards, J. Symbol. Logic 64 (1999) 1719–1742.

[4] E. Grädel, I. Walukiewicz, Guarded fixed point logic, in: Proc. 14th IEEE Symp. on Logic in Computer Science, 1999, pp. 45–54.

[5] Y. Gurevich, L. Harrington, Automata, trees and games, in: Proc. 14th. Ann. ACM Symp. on the Theory of Computing, 1982, pp. 60–65.

[6] I. Hodkinson, Loosely guarded fragment has finite model property, Studia Logica 70 (2002) 205–240.

[7] D. Kozen, Results on the propositional μ-calculus, Theoret. Comput. Sci. 27 (1983) 333–354.

[8] A. Mostowski, Games with forbidden positions, Technical Report, University of Gdańsk, 1991.

[9] D. Niwiński, Fixed points vs. infinte regeneration, Proc. 3rd IEEE, LICS, 1988, pp. 402–409.

[10] S. Pinter, P. Wolper, A temporal logic for reasoning about partially ordered computations, Proc. 3rd ACM Symp. on Principles of Distributed Computing, Vancouver, August 1984, pp. 28–37.

[11] M.O. Rabin, Decidability of second-order theories and automata on infinite trees, Trans. Amer. Math. Soc. 141 (1969) 1–35.

[12] M.O. Rabin, Weakly definable relations and special automata, Math. Logic and Foundations of Set Theory (1970) 1–23.

[13] G. Slutzki, Alternating tree automata, Theoret. Comput. Sci. 41 (1985) 305–318.

[14] M. Vardi, Reasoning about the past with two-way automata, in: Lecture Notes in Computer Science, Vol. 1443, Springer, Berlin, 1988, pp. 628–641

[15] W. Zielonka, Infinite games on finitely coloured graphs with applications to automata on infinite trees, Theoret. Comput. Sci. 200 (1998) 135–183.