

The Finite Graph Problem for Two-Way Alternating Automata

Mikołaj Bojańczyk¹

Uniwersytet Warszawski, Wydział MIM, Banacha 2, Warszawa, Polska
bojan@mimuw.edu.pl

Abstract. Two-way alternating automata on trees were introduced by Vardi [Va98]. 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.

1 Introduction

In this paper we consider the propositional μ -calculus with *backwards modalities*. It is an extension of the propositional μ -calculus introduced by Kozen [Ko83], which in itself is a very strong logic, subsuming such formalisms as Program Decision Logic, various temporal logics such as CTL and process logics such as YAPL. The various propositional μ -calculi are subject to much research, because, while being expressive, they still have reasonable computational complexity.

The propositional μ -calculus extends propositional logic with least and greatest fix-point operators μ , ν and modal quantification \exists , \forall (sometimes written as \diamond , \square). The calculus with *backwards modalities*, aside from the above, allows for quantification over backward modalities, denoted by \exists^- and \forall^- . Thus, for example, a formula of the form $\exists^- \phi$ (respectively $\exists \phi$) states that ϕ occurs in some predecessor (respectively successor) of the current state. This calculus is notably stronger than the propositional modal μ -calculus without backward modalities, in particular it no longer has the *finite model property*, i. e. it admits sentences which are satisfiable only in infinite structures. This gives rise to the following natural decision problem: „Is a given sentence of the propositional modal μ -calculus with backward modalities satisfiable in some *finite* structure?”

The more or less standard way to tackle this problem would be to use automata on infinite trees. Extensions of modal logic often have the *tree model* property, which states that structures can be *unraveled* into indistinguishable tree-like structures. This allows one to reap from the rich resources of automata on infinite trees. And thus, for instance, the decidability of the satisfiability problem for the propositional μ -calculus can be proven by a reduction to the emptiness problem for *alternating* automata on infinite trees with the *parity condition*, while the satisfiability problem for the μ -calculus with backward modalities can be reduced to the emptiness problem for alternating *two-way* automata on infinite trees with the parity condition [Va98].

As much as the tree model property is helpful in researching the satisfiability problem of the μ -calculus with backward modalities, things get more complicated where the finite model problem is concerned. The reason is that, unfortunately, finite models rarely turn out to be trees. On the contrary, the class of sentences satisfiable in finite tree models is a small subset of the class of sentences with arbitrary finite models, a class for which the finite model problem is quite easily decidable. 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 question of whether a given automaton accepts some finite graph, the *finite graph* problem.

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 automaton.

For the decidability proof of the finite graph problem, given an alternating two-way automaton A , we construct a nondeterministic automaton A' on trees that accepts unravelings of finite graphs accepted by 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 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, unravelings of finite graphs also have finite signatures. It also turns out that accepting a graph, perhaps infinite, of finite signature is in a way 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 bound signature tree. The automaton A' accepts trees where the signature of the automaton A is bound by the constant in the small signature theorem.

In the last sections of the paper we introduce the μ -calculus with backward modalities and, closely following a paper by Vardi [Va98], show its correspondence with alternating automata with the parity condition. We then show what fragment of the calculus corresponds to Büchi condition automata and prove the decidability of the finite model problem for this fragment. Finally, we comment on the possibility of further work regarding the so-called Guarded Fragment.

2 Games with the Parity Condition

Games with the parity condition are an important concept in the theory of infinite tree automata. In particular, the semantics of two-way alternating automata used in this paper are defined by a certain game with the parity condition.

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 sets of *positions*, the function $\Omega : V = V_0 \cup V_1 \rightarrow \{0, \dots, N\}$ 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*. We additionally assume that for every position $v \in V$, the set of outgoing edges $(v, w) \in E$ is finite.

The game is played as follows. The play starts in the vertex v_0 . Assuming the game 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, assume v_0, v_1, \dots is the infinite sequence 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, 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.

The notions of strategy and winning strategy are introduced in the usual manner. We say a strategy is *memoryless* if the choice of the next vertex depends solely upon the current vertex. A very important theorem [EJ91, Mo91], which will enable us to consider only memoryless strategies, says:

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.*

3 Two-Way Alternating Automata on Graphs

Two-way alternating automata were introduced by Vardi in [Va98] 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. Here we consider automata not on trees, as in the original paper, but on arbitrary graphs.

Given a set of states Q , we consider formulas built using the logical connectives \vee and \wedge from atoms of the form $\forall q$, i. e. from the set $\{\forall q, \forall^- q : q \in Q\}$ and $\exists q$ i. e. $\{\exists q, \exists^- q : q \in Q\}$. We will denote the set of such formulas by $Form(A)$. Moreover, we partition the set $Form(A)$ into *conjunctive* formulas $Con(A)$, i. e. either atoms from $\forall q$ or formulas of the form $\phi_1 \wedge \phi_2$ and *disjunctive* formulas $Dis(A)$, i. e. atoms from $\exists q$ and formulas of the form $\phi_1 \vee \phi_2$.

Definition 3.1 (Two-way alternating automaton) A *two-way alternating automaton* on Σ -labeled graphs is the tuple:

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

Q is a finite set of *states*, $q_0 \in Q$ is called the *starting state* and Ω is a function assigning to each state $q \in Q$ a natural number $\Omega(q)$ called the *color* or *priority* of q . The *transition function* δ is of the form $\delta : Q \times \Sigma \rightarrow \text{Form}(A)$.

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

To define the semantics of two-way alternating automata, we shall use games with the parity condition. Given a Σ -labeled graph $G = \langle V, E, e, v_0 \rangle$ and a two-way alternating automaton $A = \langle Q, q_0, \Sigma, \delta, \Omega \rangle$, we define the game $\mathbb{G}(A, G) = \langle V_0, V_1, E', v'_0, \Omega' \rangle$. The set of positions of $\mathbb{G}(A, G)$ is defined $V_0 = \text{Dis}(A) \times V$ and $V_1 = \text{Con}(A) \times V$.

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

The edges of the game are set as follows:

- For an atom $(\exists^{\pm} q, v) \in (\forall Q \times V) \cup (\exists Q \times V)$ there exists an edge to $(\delta(q, e(w)), w)$ if $(v, w)^i \in E$, where i is 1 if the quantifier is positive and -1 otherwise.
- For a non-atomic formula (ϕ, v) there exists an edge to (ϕ', v) for each sub-formula ϕ' of ϕ .

The coloring Ω' in the game $\mathbb{G}(A, G)$ is defined as follows: for $(\exists^{\pm} q, v) \in (\forall Q \times V) \cup (\exists Q \times V)$ we set $\Omega'(\exists^{\pm} q, v) = \Omega(q)$. For the remaining positions we set, say, $\max(\Omega(Q)) + 1$, so that their color is irrelevant. The starting position in $\mathbb{G}(A, G)$ is $(\delta(q_0, e(v_0)), v_0)$.

Definition 3.2 (Acceptance by the automaton) We say the automaton A *accepts* a graph G under strategy s if s is a winning strategy for player 0 in the game $\mathbb{G}(A, G)$. Such a strategy s is called *accepting*. We say A *accepts* graph G if there exists a strategy s such that A *accepts* G under s .

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

Definition 3.3 (Tree unraveling) Given a graph $G = \langle V, E, e, v_0 \rangle$, its *tree unraveling* is the graph $Un(G) = \langle V', E', e', v_0 \rangle$, 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

$E' = \{((\pi, v), (\pi, v, w))^i : (\pi, v, w) \in V', (v, w)^i \in E\}$ and the labeling is set as $e'(\pi, v) = e(v)$.

Note that this is a *two-way* tree, that is, edges between a son and father can be either forward or backward. The *depth* of a vertex $\pi = (v_0, \dots, v_n)$ is the number n . For two vertices π_1 and π_2 of such a tree, we say π_1 is a successor of π_2 if π_2 is an initial fragment of the path π_1 . For a tree T we use $T|_\pi$ to signify the subtree of T with the root at π , $T|_\pi^i$ is the subtree with the root at π and of depth i and $T|_\lambda^i$ is defined as $T|_\lambda^i$, where λ is the root of T .

Having a tree unraveling we define the canonical projection $\Pi : V' \rightarrow V$, so that $\Pi(\pi, v) = v$ and expand this projection onto the positions of $\mathbb{G}(A, Un(G))$ and $\mathbb{G}(A, G)$ so that $\Pi(x, (\pi, v)) = (x, v)$.

Definition 3.4 (Strategy unraveling) We say the strategy $Un(s)$ is the *unraveling of strategy s* if $\Pi \circ Un(s) = s \circ \Pi$.

We omit the trivial proof of:

Lemma 3.1. *The automaton A accepts a graph G under strategy s iff A accepts $Un(G)$ under $Un(s)$.*

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.

Example 3.0.1

Consider the following two-way automaton $A = \langle Q, q_0, \Sigma, \delta, \Omega \rangle$, where $Q = \{q_0, q_1\}$, $\Omega(q_0) = 0$, $\Omega(q_1) = 1$, and $\Sigma = \{a\}$. The transition function δ is defined as follows:

$$\begin{aligned}\delta(q_0, a) &= \exists q_0 \wedge \forall^- q_1 \\ \delta(q_1, a) &= \forall^- q_1\end{aligned}$$

We will consider a play in the game $\mathbb{G}(A, G)$ where $G = \langle \mathbb{N}, \{(n, n+1) : n \in \mathbb{N}\}, e, 0 \rangle$, such that $e(n) = a$ for all $n \in \mathbb{N}$. The play starts in formula $\exists q_0 \wedge \forall^- q_1$ at vertex 0. This is a position for player 1, let's assume he chooses the subformula $\exists q_0$ (the play stays at 0). Now player 0 has to choose a neighboring (in G) vertex along a forward edge. He has to choose 1; the position is now $\exists q_0 \wedge \forall^- q_1$ at vertex 0. This goes on until, say, we reach 10. Now let's assume player 1 chooses the subformula $\forall^- q_1$. Now it is his choice to choose a neighboring vertex in G , along a backward edge; he has to choose vertex 9. The play then goes on through positions $(\forall^- q_1, 9), \dots, (\forall^- q_1, 0)$ in which last position player 1 loses for a lack of possible moves. Consider now a different play – player 1 always chooses the subformula $\exists q_0$. The play goes through positions $(\exists q_0 \wedge \forall^- q_1, 0), (\exists q_0, 0), \dots, (\exists q_0 \wedge \forall^- q_1, k), (\exists q_0, k), \dots$. The only color appearing infinitely often in this play is 0, thus player 0 wins.

Analyzing the game $\mathbb{G}(A, G)$, one will notice that in the graph G , player 0 has a winning strategy. It can also be proven, that the automaton A accepts only graphs with an infinite forward path where no infinite backward path is ever reachable. In particular A accepts only infinite graphs.

3.1 Automaton Paths

Let us fix a two-way alternating automaton A . Given a play r in the game $\mathbb{G}(A, G)$ we can 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 $\tilde{r} = (q_0, 0), (q_1, 1), \dots, (q_0, 10), (q_1, 9), (q_1, 8), \dots, (q_1, 0)$. The following is a key definition:

Definition 3.5 (Automaton path) Let r be a play consistent with the strategy s in $\mathbb{G}(A, G)$ and let $\tilde{r} = (q_1, v_1), \dots$ be the projection of r onto $Q \times V$. Any contiguous subsequence of $\pi(r)$ is called an *automaton path* G consistent with the strategy s , or, more concisely, an automaton path in G, s .

We use $\omega(G, s)$ 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 is denoted by $|\omega|$. Given a path $\omega = (q_1, v_1), \dots, (q_n, v_n)$, we say the path $\omega' = (q_2, v_2), \dots, (q_{n-1}, v_{n-1})$ leads from (q_1, v_1) to (q_n, v_n) , which is written as $(q_1, v_1) \rightarrow^{\omega'} (q_n, v_n)$. By ω_i we denote the i -th element of the path ω , that is (q_i, v_i) . Sometimes we shall simply write ω_i to denote simply either q_i or v_i , where the context clearly defines what type of result is needed. A path ω is a *sub-path* of ω' , written as $\omega \sqsubseteq \omega'$, if ω is a contiguous subsequence of ω' . We define $\|\omega\|_Q$ as the set of states visited in ω and $\|\omega\|_V$ as the set of vertices visited in ω .

We say that a finite path ω ends well under the strategy s if it corresponds to a finite play f in $\mathbb{G}(A, G)$ winning for 0, that is one where player 1 cannot make a move. The following lemma gives a path characterization of the acceptance of automata:

Lemma 3.2. *The automaton 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 $(q_0, v_0), (q_1, v_1), \dots$ satisfies the parity condition for the sequence $\Omega(q_0), \Omega(q_1), \dots$*

Corollary 3.1. *If the automaton A accepts the graph G under s there is no cycle ω in which the lowest priority in the set $\Omega(\|\omega\|_Q)$ is odd.*

4 The Finite Graph Problem

The example in Section 3.0.1 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 are not able to prove the decidability of this full problem and we consider a simpler case. Fix a set of states Q and some subset $F \subseteq Q$. We say that the sequence of states $\{q_i\}_{i \in \mathbb{N}}$, $q_i \in Q$ satisfies the *Büchi acceptance condition*, if there exists a state $q \in F$ which appears infinitely often in the sequence $\{q_i\}_{i \in \mathbb{N}}$. This condition is obviously equivalent to the parity condition where we put $\Omega(q) = 0$ for $q \in F$ and $\Omega(q) = 1$ for $q \in Q/F$. Note that the example automaton in the previous section is an automaton with the Büchi acceptance condition. It is thus meaningful to consider the *FIN-ALT* problem restricted to automata with the Büchi acceptance condition; we shall call this problem *FIN-ALT(B)*. Consider a graph $G = \langle V, E, e, v_0 \rangle$ accepted by A . We shall now define the concept of an automaton *signature*, used in the key Theorem 4.1 of this paper.

Definition 4.1 (Signature) Let $v \in G$, $\omega \in \omega(G, s)$, and $k, i \in \mathbb{N}$.

- $Sig(\omega, i) \equiv_{def} \min\{j : \Omega(\omega_{i+j}) = 0\}$
- $Sig^{G,s}(q, v) \equiv_{def} \max\{Sig(\omega, i) : \omega \in \omega(G, s), \omega_i = (q, v)\}$.
- $Sig^{G,s}(v) \equiv_{def} (Sig^{G,s}(q, v))_{q \in Q}$

If the context is clear as to what graph and strategy are concerned, instead of $Sig^{G,s}(q, v)$ we shall write simply $Sig(q, v)$. Intuitively, $Sig(q, v)$ gives the longest possible length of an automaton path consisting of odd states starting in (q, v) . We shall assume $Sig(q, v) = \infty$ if there is no such bound.

The following theorem is the main technical result of this paper.

Theorem 4.1 (Small signature theorem). *For any alternating two-way automaton with the Büchi acceptance condition $A = \langle Q, q_0, \Sigma, \delta, \Omega \rangle$ there exists a constant M doubly exponential on $|Q|$ such that the following three conditions are equivalent:*

1. *There exists a finite graph G such that A accepts G .*
2. *There exist a bound $N \in \mathbb{N}$, a tree T and an accepting strategy s such that for every vertex $v \in T$ and every state $q \in Q$, $Sig^{T,s}(q, v) < N$.*
3. *There exist a tree T and an accepting strategy s , such that for every vertex $v \in T$ and every state $q \in Q$, $Sig^{T,s}(q, v) < M$.*

Let us fix the automaton $A = \langle Q, q_0, \Sigma, \Omega \rangle$. The proof of this theorem is long and will be distributed across three subsections.

4.1 Proof of 1 \Rightarrow 2

First we shall state two lemmas, whose trivial proofs will be omitted.

Lemma 4.1. *If the automaton A accepts the finite graph $G = \langle V, E, e, v_0 \rangle$ under s , then for every $q \in Q, v \in V$ we have $Sig^{G,s}(q, v) \leq |V||Q|$.*

Lemma 4.2. *The tree unwinding does not increase the signature, i. e.*

$$\text{Sig}^{Un(G), Un(s)}(q, (\pi v)) \leq \text{Sig}^{G, s}(q, v)$$

For the proof of $1 \Rightarrow 2$, assume A accepts the graph G . Then A accepts $Un(G)$ under the strategy $Un(s)$ (Lemma 3.1). Moreover, for every vertex πv of the tree $Un(G)$ and every state $q \in Q$ we have $\text{Sig}^{Un(G), Un(s)}(q, \pi v) \leq \text{Sig}^{G, s}(q, v) \leq |V||Q|$. The first inequality is due to Lemma 4.2, the second due to 4.1.

4.2 Proof of $3 \Rightarrow 1$

It can be shown that for each graph G accepted by A , one can reduce the edge set of G , obtaining a graph G' also accepted by A , where the degree of each vertex is bounded by a certain constant \hat{A} , linear with respect to the size of the formulas in the transition function of A . We will now try to think of a way to encode strategies. Generally speaking, an accepting strategy can choose potentially any vertex for existential atoms, making a bounded representation difficult. Fortunately however, one can show that in graphs of degree bounded by \hat{A} a strategy can be encoded as a number from 1 to a constant \tilde{A} , where \tilde{A} is exponential over \hat{A} .

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

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

It is an easy exercise to show that the function $\text{Sig}^{T, s}$ satisfies the conditions in the below lemma for the tree T' , thus proving that A accepts the finite graph T' .

Lemma 4.3. *A accepts a finite graph G under the strategy s iff there exists a number $N \in \mathbb{N}$ and a function $\sigma : Q \times V \rightarrow \{0, \dots, N\}$ such that if $(q, v)(q', v') \in \omega(G, s)$ then $\sigma(q', v') \leq \sigma(q, v)$ and the inequality is proper if $\Omega(q) = 1$.*

Proof. For the left to right implication it is sufficient to notice that the signature Sig satisfies the above conditions. For the other direction, one has to show that the function σ majorizes Sig , so that each state with odd priority appears at most N times before an even priority state appears. \square

4.3 Proof of 2 \Rightarrow 3

Let T, s be as in condition 2 of Theorem 4.1. By assumption we know there exists a certain, if perhaps difficult to estimate, bound N on the signature. We will now modify the tree T in such a way as to bound the signature by a tractable constant M .

First, we shall introduce the following definitions. We say a path $\omega \in \omega(G, s)$ is *bad* iff $\Omega(|\omega|_Q) = \{1\}$. We say (q, w) is *1-accessible* from (q', v) in G, s , written as $(q', v) \rightarrow_1 (q, w)$ in G, s , if there exists a bad path $\omega \in \omega(G, s)$ such that $(q', v) \xrightarrow{\omega} (q, w)$. Moreover, 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 *bad neighborhood* $X_{D,s}(v)$ of a vertex v in the tree D under the strategy s as the set of v 's 1-accessible (both ways) successors, i. e. $X_{D,s}(v) \equiv_{def} \{w \in D|_v : w \rightarrow_1 v \vee v \rightarrow_1 w \text{ in } D, s\}$

Let M' be a constant whose exact size depending on the size of the automaton 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 (*):

$$(*) X_{D,s}(v) \subseteq D|_v^{M'}$$

We will now construct a sequence of trees and accepting strategies $(D^0, s^0), (D^1, s^1), \dots$ such that each two tree-strategy pairs (D^i, s^i) and (D^j, s^j) are identical up to depth $\min(i, j) - 1$ and, moreover, if a vertex v of the tree D^i has a depth less than i , then the property (*) holds for v in D^i, s^i .

We will define the trees inductively with respect to i . Let D^0, s^0 be simply T^0, s^0 . The above condition obviously holds 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 4.4 for successive vertices of depth i . Of course the conditions in the lemma are satisfied by D^0, s^0 .

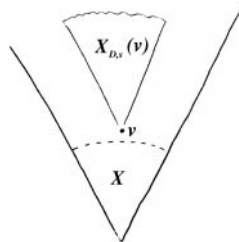


Fig. 1. The tree D

Lemma 4.4. *Let D, s be such that*

1. A accepts D under s
2. $X_{D,s}(w)$ is finite for every vertex w of the tree D .
3. (*) holds for some contiguous subset of vertices X closed under the ancestor relation.

Let v be the successor of any vertex in X . There exist a tree D' and strategy s' identical with D, s on the set $X \cup \{v\}$ such that the above three conditions hold for D', s' and $(*)$ holds for v .

Proof. We prove 4.4 by iterating Lemma 4.5. □

Lemma 4.5. Assume D, s and v satisfy the assumptions of Lemma 4.4. If $(*)$ does not hold for v , there exists a tree D' and strategy s' , identical with D, s on the set $X \cup \{v\}$ such that D', s' satisfies the assumptions of Lemma 4.4 and moreover the following inequality holds (the cardinality of both sets is finite by assumption 3):

$$|X_{D',s'}(v)| < |X_{D,s}(v)|$$

Proof. We will consider a certain equivalence relation \simeq defined on $X_{D,s}(v)$. Vertices equivalent under this relation are in a sense interchangeable (along with their subtrees). First, introduce for any two vertices w, w' in the tree D the following symbol $O(w, w') \subseteq Q \times P(\{0, 1\}) \times Q$. Let $O(w, w') = \{(q, R, q') : \exists \omega \in \omega(D, s). (q, w) \xrightarrow{\omega} (q', w') \wedge \Omega(\|\omega\|_Q) = R\}$. Using this notation, we write $w \simeq w'$ if and only if 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. $O(w, v) = O(w', v)$

Let M' be the number of abstraction classes of the relation \simeq . It is easy to see that the bound M' is exponential with respect to $|Q|$. Assume now that the depth of $X_{D,s}(v)$ is greater than M' . In such a case we can find two vertices $w, w' \in X_{D,s}(v), w < w'$ such that $w \simeq w'$. Now take for D' (from Lemma 4.5) the tree resulting from substituting $D|_w$ for $D|_{w'}$, and let s' be the strategy s restricted to the new, smaller tree.

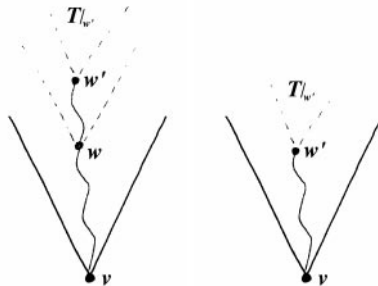


Fig. 2. Before and after the cut ($D|_v$ and $D'|_v$)

Definition 4.2 (Clean path) We say the path ω is *clean*, if $w' \notin \|\omega\|_V$. Clean paths can be either *upper*, that is contained in $D'|_{w'}$, or *lower* – the remainder.

Fact 4.1 For any state $q \in Q$, if (q, w') is accessible in D', s' , then both (q, w) and (q, w') are accessible in D, s .

Proof. Let ω be such that $(q_0, v_0) \xrightarrow{\omega} (q, w')$ in D', s' , where v_0 is the root and starting vertex of D' . The proof is by induction on the number of times w' appears in ω . For clean paths this is obvious, using condition 3 of $w \simeq w'$. Assume ω is not clean. Then $\omega = \omega^1(q', w')\omega^2$, where ω^2 is clean. Let us just consider the case where ω^2 is upper, the proof of the other is analogous. If ω^2 is upper then it is a correct path in D, s such that $(q', w') \xrightarrow{\omega^2} (q, w')$ in D, s . By condition 2 of $w \simeq w'$ we have $(q', w) \xrightarrow{\omega^3} (q, w)$ in D, s for some path ω^3 . By induction hypothesis, both (q', w') and (q', w) are accessible in D, s , say by paths ω^A and ω^B . Linking the paths ω^A and ω^B with ω^2 and ω^3 respectively we obtain the desired assertion. \square

Corollary 4.1. For any $q \in Q, u \in D'$, if (q, u) is accessible in D', s' then it is also accessible in D, s .

Now we are in a position to prove Lemma 4.5. That A accepts the tree D' under s is a rather tedious technical result whose proof we omit here. Secondly, we want to show that $X_{D', s'}(u)$ is finite for each vertex u in the tree D', s' . Take any bad path ω that goes through u and consider two cases:

- ω is clean. Using Corollary 4.1, we show $\omega \in \omega(D, s)$, so we can, according to the assumptions of Lemma 4.5, bound the depth of ω by a constant dependent only on the vertex u .
- ω is not clean. Again – divide ω into clean sub-paths:

$$\omega = \omega^{(1)}(w', q^{(1)})\omega^{(2)} \dots (w', q^{(n-1)})\omega^{(n)}$$

By easy induction we can prove that if $\omega^{(i)}$ upper, then $w' \rightarrow_1 \omega^{(i)}$ w D, s . This proves that the depth of the path ω is no greater than the depth of $X_{D, s}(w')$, in particular it is bound.

Finally, the facts that the cut preserves the property (*) for vertices $u \in X$ and that $|X_{D', s'}(v)| < |X_{D, s}(u)|$ are implied by the following fact:

Fact 4.2 If $u \in X \cup \{v\}$, then $X_{D', s'}(u) \subseteq X_{D, s}(u) \cap D'$

To prove Fact 4.2, we need to prove the following lemma:

Lemma 4.6. If $(q, v) \rightarrow_1 (q', w')$ [respectively $(q', w') \rightarrow_1 (q, v)$] in D', s' , then we have both $(q, v) \rightarrow_1 (q', w)$ and $(q, v) \rightarrow_1 (q', w')$ [$(q', w') \rightarrow_1 (q, v)$ and $(q', w) \rightarrow_1 (q, v)$] in D, s .

The proof is wholly analogous to the proof of Lemma 4.1 and will be omitted. Thus equipped we can finish the proof of Fact 4.2 and along with that, the whole Lemma 4.5.

Let $u \in X \cup \{v\}$ and $u' \in X_{D',s'}(u)$. We want to show that $u' \in X_{D,s}(u) \cap D'$. We will only consider the first case in the definition of $X_{D',s'}(u)$, the other one is similar. Assume then that $u \rightarrow_1 u'$. There exists states $q, q' \in Q$ and a bad path $\omega \in \omega(D', s')$ such that $(q, u) \xrightarrow{\omega} (q', u')$. If ω is clean, then obviously $(q, u) \xrightarrow{\omega} (q', u')$ holds also in D, s and so $u' \in X_{D,s}(u) \cap D'$. In the other case, let $\omega = \omega_1(q'', w')\omega_2$, where ω_2 is clean. Consider two cases:

- ω_2 is upper. According to Lemma 4.6, we have $(q, u) \rightarrow_1 (q'', w')$ in D, s . Since the path ω_2 is a correct path in D, s , we have $(q, u) \rightarrow_1 (q', u')$ in D', s' and, consequently, $u' \in X_{D,s}(u) \cap D'$.
- ω_2 is lower. Analogously to the above, except using $(q, u) \rightarrow_1 (q', w)$.

□

(end of proof of 4.5)

Having thus proven 4.5 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 $X_{D^n, s^n}(v)$ contains only vertices of depth less than n , we see that $X_{D,s}(v) \subseteq D|_v^{M'}$, or, in other words, (*) holds for all all vertices of D, s . It is now a trivial observation that the maximal length M of a bad path contained within $D|_v^{M'}$ is at most exponential with respect to M' , otherwise we would have a cycle.

4.4 The FIN-ALT(B) Problem Is Decidable

Armed with Theorem 4.1 we are ready to show the decidability of the FIN-ALT(B) problem.

Theorem 4.2. *The FIN-ALT(B) problem is decidable in DEXPTIME.*

Proof. Given an automaton A we shall construct a nondeterministic automaton A' on full \hat{A} -ary trees over the alphabet $\Sigma \times \{1, \dots, \hat{A}\}$. The automaton A' will recognize an infinite tree T labeled by the strategy s iff for every vertex v of the tree T , the signature $Sig^{T,s}$ is smaller than the constant M from Theorem 4.1. The automaton A' guesses a function $\sigma : V \times Q \rightarrow \{1, \dots, M\}$ defined on vertices of the tree T and then checks if the assumptions of Lemma 4.3 hold.

This kind of local property can be checked by a nondeterministic automaton whose number of states is doubly exponential with regards the size of Q of the automaton A (this is how big the constant M is). The fact that A' works on \hat{A} -ary trees is not limiting, as we have observed before. Thanks to Theorem 4.1, we need only look for trees whose signature is bounded by M , so the emptiness problem for A' (in terms of nondeterministic automata) is equivalent to the emptiness of A (in terms of two-way alternating automata).

Since the automaton A' checks only a local consistency, it has the following nice property: if A' has an infinite run, then it has an accepting run. It can be proved that for such automata the emptiness problem is decidable in polynomial time and thus we obtain the time in the theorem's conclusion. □

5 The Propositional 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 5.1 (Formulas of the calculus) The set of formulas of the μ -calculus is the smallest set such that:

- Every atomic proposition $p \in AP$ 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:

$$\alpha \vee \beta, \alpha \wedge \beta, \exists \alpha, \forall \alpha, \exists^- \alpha, \forall^- \alpha, \mu X. \alpha, \nu X. \alpha$$

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 5.2 (Kripke structure) A *Kripke structure* $K = \langle V, E, S \rangle$ consists of a graph $\langle V, E \rangle$ along with a function $S : V \rightarrow P(AP)$ 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 $v : VAR \rightarrow P(V)$. 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 $X \in VAR$, $X^K[v] = v(X)$
- $(\phi_1 \wedge \phi_2)^K[v] = \phi_1^K[v] \cap \phi_2^K[v]$
- $(\phi_1 \vee \phi_2)^K[v] = \phi_1^K[v] \cup \phi_2^K[v]$
- $(\exists^k \phi)^K[v] = \{u \in V : \exists w \in V. (u, w)^k \in E \wedge w \in \phi^K[v]\}, k \in \{1, -1\}$
- $(\forall^k \phi)^K[v] = \{u \in V : \forall w \in V. (u, w)^k \in E \Rightarrow w \in \phi^K[v]\}, k \in \{1, -1\}$
- $(\mu X. \phi)^K[v] = \bigcap \{V' \subseteq V : \phi^K[v[V'/X]] \subseteq V'\}$
- $(\nu X. \phi)^K[v] = \bigcup \{V' \subseteq V : V' \subseteq \phi^K[v[V'/X]]\}$

5.1 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 $\{\circ q : q \in Q\}$. 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 loses; 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 . We omit the easy, yet technical, proof of:

Lemma 5.1. *Every enhanced automaton is equivalent to a two-way alternating automaton with only a polynomial blowup of size. The finite graph problem for enhanced automata with the Büchi condition is decidable in double exponential time with respect to the size of states.*

5.2 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 5.3 (Encoding) The *encoding* of a Kripke structure $K = \langle V, E, S \rangle$ from vertex $v_0 \in V$ is the graph $G(K, v_0) = \langle V, E, e, v_0 \rangle$ where $e : V \rightarrow \Sigma_\phi$ is the restriction of S to Σ_ϕ .

Let ϕ be a sentence of the μ -calculus. We will construct an enhanced automaton A_ϕ 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 $A_\phi = \langle cl(\phi), \phi, \Sigma_\phi, \delta, \Omega \rangle$. The transition function δ is defined as follows:

- $\delta(p, \sigma) = \text{true}$ if $p \in \sigma$, *false* otherwise.
- $\delta(\neg p, \sigma) = \text{false}$ 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 5.2. [Va98] *For every sentence ϕ and every Kripke structure K , $v_0 \in \phi^K$ if and only if A_ϕ accepts the encoding $G(K, v_0)$.*

Corollary 5.1. (of Lemma 5.2) *A sentence ϕ of the μ -calculus has a finite model if and only if the automaton A_ϕ accepts some finite graph.*

We say a sentence of the μ -calculus is *simple*, 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 5.2, simple 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 5.1. *The problem whether a simple sentence of the μ -calculus with backward modalities has a finite model is in DEXPTIME.*

Proof. Given a simple sentence ϕ we construct an equivalent automaton A'_ϕ and solve this instance of the enhanced *FIN-ALT(B)* problem. Both the alphabet and set of states of the automaton A'_ϕ are polynomial with respect to the length of the formula ϕ . \square

6 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 4.1, 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 the paper [GW99] to decide the satisfiability of formulas of the so-called Guarded Fragment with fixed points.

The Guarded Fragment [ABN98] is a subset of first order logic, originally introduced as an elaboration on the translation of modal logic into first order logic. Currently the Guarded Fragment is subject to much research. It can be supposed that the *FIN-ALT* problem 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

- [ABN98] H. Andreka, J. van Benthem and I. Nemeti, Modal Languages and Bounded Fragments of Predicate Logic, *Journal of Philosophical Logic*, 27 (1998), pp. 217-274.
- [EJ91] E. A. Emerson and C. Jutla: Tree Automata, Mu-Calculus and Determinacy, in *Proc. 32th IEEE Symposium on Foundations of Computer Science* pages 368-377.
- [ES89] R. S. Street and E. A. Emerson: An Automata theoretic procedure for the propositional mu-calculus. *Information and Computation*, 81:249-264.
- [GW99] E. Grädel and I. Walukiewicz: Guarded Fixed Point Logic, *Proc. 14th IEEE Symp. on Logic in Computer Science*, pages 45-54.
- [Ko83] D. Kozen: Results on the Propositional μ -calculus, in *Theoretical Computer Science, Vol. 27* pages 333-354.
- [LPZ85] O. Lichtenstein, A. Pnueli and L. Zuck: The Glory of the Past, in *Logics of Programs, Vol. 193 LNCS* pages 196-218.
- [Mo91] A. W. Mostowski: Games with Forbidden Positions, Technical Report 78, University of Gdańsk, 1991.

- [MS87] D. E. Muller and P. E. Schupp: Alternating automata on infinite trees, *Theoretical Computer Science* , 54:267-276, 1987.
- [Ni88] D. Niwiński: Fixed Points vs. Infinite Generation, in *Proc. 3rd IEEE LICS* pages 402-409.
- [St82] R. S. Streett: Propositional dynamic logic of looping and converse, in *Information and Control* , Vol. 54, pages 121-141.
- [Th97] Wolfgang Thomas: Languages, Automata, and Logic, in *Handbook of Formal Language Theory, III* , Springer, pages 389-455.
- [Va97] M. Vardi: Why is modal logic so robustly decidable?, in *Descriptive Complexity and Finite Models* , AMS, pages 149-184.
- [Va98] M. Vardi: Reasoning About the Past with Two-way Automata, in *Vol. 1443 LNCS* pages 628-641.