Deciding the Topological Complexity of Büchi Languages

Michał Skrzypczak*[1] and Igor Walukiewicz[2]

1 **Institute of Informatics, University of Warsaw, Warsaw, Poland**
mskrzypczak@mimuw.edu.pl

2 **LABRI, Bordeaux, France**
igw@labri.fr

Abstract

We study the topological complexity of languages of Büchi automata on infinite binary trees. We show that such a language is either Borel and WMSO-definable, or Σ^1_1-complete and not WMSO-definable; moreover it can be algorithmically decided which of the two cases holds. The proof relies on a direct reduction to deciding the winner in a finite game with a regular winning condition.

1998 ACM Subject Classification F.1.1 Models of Computation

Keywords and phrases tree automata, non-determinism, Borel sets, topological complexity, decidability

Digital Object Identifier 10.4230/LIPIcs.ICALP.2016.99

1 Introduction

The class of regular languages of infinite trees is one of the most important classes of properties of infinite computations. Similarly to the arithmetic hierarchy, the class is structured into the so called Mostowski–Rabin index hierarchy. This hierarchy reflects the complexity of a language in terms of an alternation of fix-points needed to express it, or equivalently, in terms of the minimal complexity of the acceptance condition of an automaton accepting the language. While we know for about two decades that the hierarchy is infinite [5], we are still very far from understanding it. One important objective in this area is to effectively characterise every level of the hierarchy: for a given regular language of infinite trees calculate its level in the hierarchy.

The difficulty in understanding the Mostowski–Rabin index hierarchy of tree languages is linked to the lack of deterministic acceptors for such languages. Thus, on a smaller scale, we face here the same problem as in the complexity theory, namely the problem of understanding the structure of non-deterministic computations. When restricted to deterministic acceptors, the Mostowski–Rabin hierarchy is by now well-understood. For every level we know a pattern such that the pattern appears in a deterministic tree automaton if and only if the language recognised by this automaton is hard for this level [21, 18, 19]. The pattern method has been extended to the so called game automata [12], but there is no hope to use it for non-deterministic automata.

Apart from decidability questions, a promising way to understand the Mostowski–Rabin hierarchy is to relate it to the topological hierarchy. (For an introduction to the classes

* The first author has been supported by Poland's National Science Center grant (decision DEC-2012/07/D/ST6/02443).

43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016).
Editors: Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi;
Article No. 99; pp. 99:1–99:13
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

of topological complexity see for instance [15].) Topological properties of sets defined by automata are discussed in [24]. It is well-known that all regular languages of infinite trees are contained in the $\mathbf{\Delta}^1_2$ level of the projective hierarchy. The languages of Büchi automata, or equivalently those definable in existential MSO logic, are contained in the $\mathbf{\Sigma}^1_1$ level. The languages of weak-alternating automata, or equivalently definable in weak MSO (WMSO), are Borel; moreover for every finite level of the Borel hierarchy there is a complete weakly definable language [23]. In [23], Skurczyński asks if every regular language that is Borel is necessarily weakly definable. In this paper we answer this question for languages recognised by Büchi automata, as expressed by the main theorem.

▶ **Theorem 1.** *If $\mathcal{B}$ is a non-deterministic Büchi tree automaton then one of the following possibilities holds and it is possible to effectively decide which one it is:*
1. $\mathrm{L}(\mathcal{B})$ *is Borel and* WMSO*-definable,*
2. $\mathrm{L}(\mathcal{B})$ *is* $\mathbf{\Sigma}^1_1$*-complete and not* WMSO*-definable.*

The theorem is proved through a game construction. Given a Büchi automaton $\mathcal{B}$ we construct a finite game $\mathcal{F}(\infty)$ such that if $\exists$ wins in this game then the language of $\mathcal{B}$ is $\mathbf{\Sigma}^1_1$-complete; but if $\forall$ wins then the language of $\mathcal{B}$ can be accepted by a weak alternating automaton constructed from $\mathcal{B}$. A similar technique of relying on the finite memory determinacy of regular games was used in [1].

Related work. Colcombet, Kuperberg, Löding, and Vanden Boom [16, 8] have proved the algorithmic part of the above theorem; using some decidability result in the theory of cost functions and a reduction of Colcombet and Löding [9] they have shown how to decide if the language of a Büchi automaton is weakly definable. The topological counterpart of Theorem 1 seems not to follow from their construction. Our proof relies only on standard facts from automata theory, and may be simpler, at least for those who are not familiar with the theory of cost functions.

Finding effective characterisations of various classes of infinite tree languages is an important topic of language theory. As we noted above, for languages of deterministic tree automata the situation is quite well-understood; but for the case of all regular tree languages for some time it was only known how to decide if a given regular language can be accepted by an automaton with a trivial acceptance condition [17, 25]. The theory of cost functions allows to decide if a given language can be accepted by a non-deterministic co-Büchi automaton [8]. Bojańczyk and Idziaszek [2] have recently shown decidability of definability in a temporal logic EF. Bojańczyk and Place [4] show how to decide if a given language is a Boolean combination of open sets.

The study of topological properties of regular languages of trees has seen important advances too [10, 13, 6, 18, 19, 3, 11]. Over a decade ago an interesting gap property has been observed [20] for languages of deterministic tree automata: a language is either $\mathbf{\Pi}^1_1$-complete, or contained in the $\mathbf{\Pi}^0_3$ level of the Borel hierarchy. A similar gap property has been recently shown for languages of thin trees [14]: a regular language of thin trees, treated as a subset of all trees, is either definable in weak MSO logic or $\mathbf{\Pi}^1_1$-complete. Our theorem shows a gap property for languages of non-deterministic Büchi automata.

2 Preliminaries and an outline of the construction

We write ω for the natural numbers, and $\overline{\omega}$ for the extension of ω with a greatest element ∞. So $n < \infty$ for all $n \in \omega$, and $\infty - 1 = \infty$. We assume standard definitions of non-deterministic and alternating automata on infinite binary trees. All trees we consider are binary, the two directions are $\mathtt{L}$ and $\mathtt{R}$. The root of a tree is ϵ. By $\preceq$ we denote the prefix order on the nodes of a tree. In this paper we will use perfect information two player games of infinite duration. The players are denoted $\exists$ (Eve) and $\forall$ (Adam).

A non-deterministic Büchi automaton is a tuple $\mathcal{B} = \langle Q, A, q_{\mathrm{I}} \in Q, \delta \subseteq Q \times A \times Q \times Q, F \subseteq Q \rangle$. We will use the standard notion of a run ρ over a tree t. A run of $\mathcal{B}$ is accepting if on every branch some state from F appears infinitely often. A weak alternating automaton is very similar but for a total quasi-order on states, and the transition relation that now sends a non-empty set of states in every direction $\delta \subseteq Q \times A \times \mathsf{P}^+(Q) \times \mathsf{P}^+(Q)$ (by $\mathsf{P}^+(X)$ we denote the set of non-empty subsets of X). The transition relation should respect the order on the states in a sense that if $(q, a, S_{\mathtt{L}}, S_{\mathtt{R}}) \in \delta$ then all the states in $S_{\mathtt{L}} \cup S_{\mathtt{R}}$ should not be bigger than q, and if $q \in F$ is accepting then all not accepting states in $S_{\mathtt{L}} \cup S_{\mathtt{R}}$ should be strictly smaller than q in this order. A weak alternating automaton $\mathcal{A}$ induces a game on every tree t: the positions of this game are (u, q) with $u \in \{\mathtt{L}, \mathtt{R}\}^*$ and $q \in Q$; the initial position is $(\epsilon, q_{\mathrm{I}})$; from a position (u, q) first $\exists$ chooses a transition $(q, t(u), S_{\mathtt{L}}, S_{\mathtt{R}})$, then $\forall$ chooses a direction d and a state $q' \in S_d$, the successive position is (ud, q'). A play is won by $\exists$ if it contains infinitely many accepting states. Without loss of generality we assume that all the considered automata are complete: for every state $q \in Q$ and every letter $a \in A$ there is at least one transition from q over a. For an automaton $\mathcal{A}$, by $\mathrm{L}(\mathcal{A})$ we denote the set of trees accepted by $\mathcal{A}$.

Our proof of Theorem 1 will use two games, or rather game families constructed from $\mathcal{B}$. The first game, $\mathcal{G}(t)$, will be played on a tree t. The game will encode in a compact way not only the acceptance of t by $\mathcal{B}$, but also possible approximations of $\mathcal{B}$ by weak automata. It is motivated by the technical core of the construction in [22]. More precisely, for every $K \in \overline{\omega}$ we will have a variant $\mathcal{G}(t, K)$ of the game. Each game defines a language of trees

$$\mathrm{L}(\mathcal{G}, K) = \{t \mid \exists \text{ wins } \mathcal{G}(t, K)\}. \tag{1}$$

The game $\mathcal{G}(t, \infty)$ will encode the acceptance of t by $\mathcal{B}$, i.e., $\mathrm{L}(\mathcal{G}, \infty) = \mathrm{L}(\mathcal{B})$. For every $K \in \omega$, the game $\mathcal{G}(t, K)$ will encode the acceptance of t by some specific weak alternating automaton obtained from $\mathcal{B}$; in particular $\mathrm{L}(\mathcal{G}, K)$ will be WMSO-definable. The parameter K will control the quality of the approximation of $\mathcal{B}$, in a sense that

$$\mathrm{L}(\mathcal{G}, 0) \supseteq \mathrm{L}(\mathcal{G}, 1) \supseteq \cdots \supseteq \mathrm{L}(\mathcal{B})\,.$$

We will show that $\mathrm{L}(\mathcal{B})$ is WMSO-definable if and only if $\mathrm{L}(\mathcal{B}) = \mathrm{L}(\mathcal{G}, K)$ for some finite bound $K \in \omega$. Moreover, we will show that a candidate K_0 for this bound can be computed from $\mathcal{B}$. These results will be obtained from the analysis of another game that we call $\mathcal{F}$.

The game $\mathcal{F}$, and its variants $\mathcal{F}(K)$ for all $K \in \overline{\omega}$, will be central for our arguments. For every $K \in \overline{\omega}$, the game $\mathcal{F}(K)$ will be finite in a sense that there will be a finite number of positions reachable from the initial position. The game $\mathcal{F}(K)$ will in some sense simulate $\mathcal{G}(t, K)$ for an unknown t. We will show that when $K \in \omega$ is too small for $\mathcal{B}$, i.e. when $\mathrm{L}(\mathcal{G}, K) \supsetneq \mathrm{L}(\mathcal{B})$, then $\exists$ has a winning strategy in $\mathcal{F}(K)$. Next, we examine $\mathcal{F}(\infty)$ and show that the winner in this game determines if $\mathrm{L}(\mathcal{B})$ is WMSO-definable. If $\exists$ does not win in $\mathcal{F}(\infty)$ then she does not win in $\mathcal{F}(K_0)$ for some K_0 computable from $\mathcal{B}$ (Proposition 10). Thus $\mathrm{L}(\mathcal{B}) = \mathrm{L}(\mathcal{G}, K_0)$ is WMSO-definable. The most difficult part of the proof is to show that if $\exists$ wins in $\mathcal{F}(\infty)$ then $\mathrm{L}(\mathcal{B})$ is $\mathbf{\Sigma}^1_1$-complete and thus not WMSO-definable (Proposition 11).

The way in which the game $\mathcal{F}$ is obtained from $\mathcal{G}$ is motivated by the concept of *history determinism* and in particular by the combinatorial structure of *domination games*, see [7].

3 The game $\mathcal{G}(t)$

Let us start with the game $\mathcal{G}(t)$. The positions of $\mathcal{G}(t)$ are of the form (q, u, K, z) where $q \in Q^{\mathcal{B}}$ is a state, $u \in \{\mathtt{L}, \mathtt{R}\}^*$ is a node of t, $K \in \overline{\omega}$ is a counter value, and z is one of the three special symbols: *choice*, *safe*, or *reach*. The z component determines the possible choices from a position (q, u, K, z):

$z = \textit{choice}$: In this case $\forall$ chooses $z' \in \{\textit{safe}, \textit{reach}\}$. If he chooses $z' = \textit{safe}$ then $K' = K$, if he chooses $z' = \textit{reach}$ then $K' = K - 1$; in particular if $K = 0$ then $\forall$ has to choose $z' = \textit{safe}$. The game proceeds to the position (q, u, K', z').

$z = \textit{safe}$: First $\exists$ proposes a transition of $\mathcal{B}$ of the form $(q, t(u), q_{\mathtt{L}}, q_{\mathtt{R}})$ and then $\forall$ chooses a direction $d \in \{\mathtt{L}, \mathtt{R}\}$. The game proceeds to the position $(q_d, ud, K, \textit{choice})$.

$z = \textit{reach}$: First $\exists$ proposes a transition of $\mathcal{B}$ of the form $(q, t(u), q_{\mathtt{L}}, q_{\mathtt{R}})$ and then $\forall$ chooses a direction $d \in \{\mathtt{L}, \mathtt{R}\}$. If q_d is an accepting state then the game proceeds to the position $(q_d, ud, K, \textit{choice})$, otherwise it proceeds to the position $(q_d, ud, K, \textit{reach})$.

Every play of $\mathcal{G}(t)$ is infinite. Such a play is won by $\forall$ if $z = \textit{reach}$ from some point on. In the opposite case (i.e. if $z = \textit{choice}$ infinitely many times) $\exists$ wins.

As we can see from the definition, the game proceeds in phases. It is $\forall$ who chooses if the game should be in a *safe* phase or in a *reach* phase. In the *safe* phase players just construct a path from a run of $\mathcal{B}$ ignoring the acceptance condition. In the *reach* phase $\exists$ needs to provide a finite part of a run until the next accepting states. The counter K gives a bound on the number of *reach* phases: each time $\forall$ chooses *reach*, K is decreased. Notice that if K is ∞ then it stays ∞ during the whole play, so there can be infinitely many *reach* phases.

For $K \in \overline{\omega}$ we denote by $\mathcal{G}(t, K)$ the game $\mathcal{G}(t)$ with the initial position $(q_{\mathrm{I}}^{\mathcal{B}}, \epsilon, K, \textit{choice})$.

▶ **Example 2.** For our running example we consider trees over the alphabet $\{a, b\}$ and a Büchi automaton $\mathcal{B}$ accepting the trees with a branch having infinitely many occurrences of a. This automaton has three states $q_a, q_b, \top$, with both q_a, and $\top$ accepting. For a state q_x the transition relation $\delta^{\mathcal{B}}$ on a letter y contains the transitions $(q_x, y, q_y, \top)$ and $(q_x, y, \top, q_y)$; for $x, y \in \{a, b\}$. From $\top$ the automaton stays in $\top$ on every letter and in every direction.

Using the notation from (1) we can see that for $K = 0$ the language $L(\mathcal{G}, K)$ is simply the language of all trees. For $K > 0$, it is the language of trees having a path such that every node on this path has a descendant whose label is a and the subtree rooted in this descendant is in $\mathrm{L}(\mathcal{G}, K - 1)$.

The next three lemmas give connections between the game $\mathcal{G}(t)$ and the automaton $\mathcal{B}$. They refer to the languages $\mathrm{L}(\mathcal{G}, K)$ of the game as defined in (1).

▶ **Lemma 3.** $\mathrm{L}(\mathcal{B}) = \mathrm{L}(\mathcal{G}, \infty)$.

▶ **Lemma 4.** *If $\exists$ wins $\mathcal{G}(t)$ from a position (q, u, K, z) and $K' \leq K \in \overline{\omega}$ then $\exists$ wins $\mathcal{G}(t)$ from the position (q, u, K', z). In other words $\mathrm{L}(\mathcal{G}, K') \supseteq \mathrm{L}(\mathcal{G}, K) \supseteq \mathrm{L}(\mathcal{G}, \infty)$.*

▶ **Lemma 5.** *For every $K \in \omega$ the set of trees $\mathrm{L}(\mathcal{G}, K)$ can be recognised by a weak alternating automaton (or equivalently, it is wmso-definable).*

4 The game $\mathcal{F}$

We proceed to a definition of the game $\mathcal{F}$, and its variants $\mathcal{F}(K)$, for $K \in \overline{\omega}$. For all $K \in \overline{\omega}$, the game $\mathcal{F}(K)$ will simulate $\mathcal{G}(t, K)$ with an unknown t generated *on-the-fly*.

Let us fix a non-deterministic parity tree automaton $\mathcal{A}$ recognising the complement of $\mathrm{L}(\mathcal{B})$ ($\mathcal{A}$ may not be equivalent to a Büchi automaton). We will construct $\mathcal{F}$ from $\mathcal{A}$ and $\mathcal{B}$. Intuitively, in $\mathcal{F}$ we ask the players to proceed as follows:

- $\exists$ should successively construct a tree t and a run $\rho^{\mathcal{A}}$ of $\mathcal{A}$ on t;
- at the same time $\forall$ should select directions in this tree constructing an infinite branch α of t, aiming to show that the run $\rho^{\mathcal{A}}$ proposed by $\exists$ is not accepting;
- moreover $\exists$ would aim at showing that the tree t she constructs is difficult in a sense that she can win $\mathcal{G}(t, K)$ for all finite K. $\forall$ will aim to refute this claim, by showing that he can win with some finite K. It is crucial for our argument that $\forall$ can do this in a *history-deterministic* way in the sense of [7].

4.1 Positions and multi-transitions

The positions of $\mathcal{F}$ are of the form (S, p, κ, r) where:
- $S \in \mathsf{P}(Q^{\mathcal{B}} \times \{\mathit{safe}, \mathit{reach}\})$ is a set of *active states*,
- $p \in Q^{\mathcal{A}}$ is a state of the automaton $\mathcal{A}$,
- $\kappa \colon S \to \overline{\omega}$ assigns to the active states their *counter values*,
- $r \in \{0, 1, 2\}$ is a *sub-round number*.

Using the first and the third component $\exists$ will try to prove that she wins in all the games $\mathcal{G}(t, K)$. In the second component she will construct a run of $\mathcal{A}$. The fourth component makes the definition of the game more modular.

Similarly as before, for $K \in \overline{\omega}$ by $\mathcal{F}(K)$ we denote the game $\mathcal{F}$ with the initial position $(\{(q_{\mathrm{I}}^{\mathcal{B}}, \mathit{safe})\}, q_{\mathrm{I}}^{\mathcal{A}}, \kappa, 0)$ where $\kappa(q_{\mathrm{I}}^{\mathcal{B}}, \mathit{safe}) = K$.

We say that an active state (q, z) is *in the safe phase* if $z = \mathit{safe}$; and *in the reach phase* if $z = \mathit{reach}$. A pair (s, s') *changes phases* if s and s' are in different phases. So (s, s') can *change phases from safe to reach*, or *change phases from reach to safe*.

The edges of $\mathcal{F}$ will have an additional structure (i.e. an edge will be more than just a pair of positions of the game). This richer structure will be used to define the winning condition of $\mathcal{F}$ that will refer to a sequence of edges. From our definition it will be easy to see how to transform such a game into a standard two player game. To underline that edges have additional structure we refer to them as *multi-transitions*.

A *multi-transition* μ from a position (S, p, κ, r) to a position (S', p', κ', r') contains:
- the *pre-state* (S, p, κ, r),
- the *post-state* (S', p', κ', r') with $r' = r + 1 \mod 3$,
- a set $e \subseteq S \times S'$ of *edges* from the active states in S to the active states in S',
- a set $\bar{e} \subseteq e$ of *boldfaced edges*, with exactly one boldfaced edge leading to every $s' \in S'$:

$$\forall_{s' \in S'} \ \big|\{s : (s, s') \in \bar{e}\}\big| = 1 \tag{2}$$

We additionally require the following condition on κ, called *boldface-decreasing*. Assume that $(s, s') \in \bar{e}$. If (s, s') changes phases from *safe* to *reach* then[1] $\kappa'(s') = \max(\kappa(s)-1, 0)$. Otherwise $\kappa'(s') = \kappa(s)$.

An example multi-transition is depicted in Figure 1. The role of e is to trace the origins of each active state in a similar way as for determinisation of Büchi automata. With the boldfaced edges $\forall$ will indicate which of the possible origins of an active state he finds the most promising for him. The *boldface-decreasing* condition says that on boldfaced traces the counter should behave in the same way as in a game $\mathcal{G}(t, K)$ for the tree t being constructed. The exact rules how the multi-transitions are selected by the players are given in Section 4.2.

[1] By the rules of the game, we shall never have $\kappa(s) < 1$ in this case.

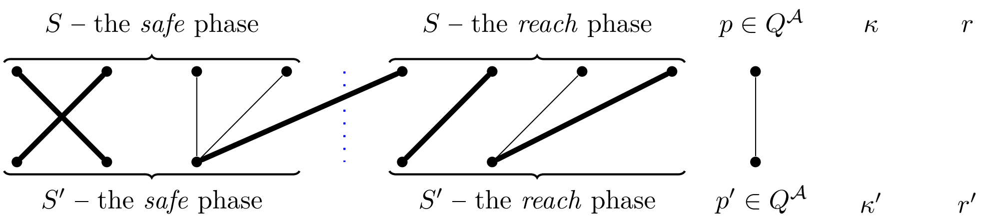

Figure 1 An example multi-transition μ. The big dots correspond to the active states and the state of $\mathcal{A}$. The convention is that all the active states from the *safe* phase are drawn on the left, all the active states from the *reach* phase are drawn in the middle, then comes the state of $\mathcal{A}$, and finally the counter values κ and the sub-round number r are drawn on the right. For the purpose of the layout, we additionally draw an edge between the states p and p' of $\mathcal{A}$ (this edge does not belong to e). The dotted line separates the *safe* phase from the *reach* phase. Boldfaced edges are boldfaced. Every active state in S' has one incoming boldfaced edge as required by (2).

Notice that for a fixed $K \in \overline{\omega}$ the game $\mathcal{F}(K)$ has only finitely many reachable positions and finitely many multi-transitions – if $K = \infty$ then all the counter values κ are equal ∞, otherwise the counter values in the reachable positions are at most K.

4.2 Rules of the game $\mathcal{F}$

In this section we describe the rules of the game $\mathcal{F}$. From a position (S, p, κ, r) the players interact constructing a new position (S', p', κ', r') and a multi-transition between the two positions. For this they select a set of edges $e \subseteq S \times (Q^{\mathcal{B}} \times \{\mathit{safe}, \mathit{reach}\})$ and a state $p' \in Q^{\mathcal{A}}$ according to the rules given below. Then $\forall$ chooses an arbitrary multi-transition μ that *respects* (S, p, κ, r)*,* e*, and* p' in the following sense:

- the pre-state of μ is (S, p, κ, r),
- the post-state of μ is (S', p', κ', r'); where $S' = \{s' : (s, s') \in e\}$ consists of the targets of the edges e, κ' is determined by the boldface-decreasing condition, and $r' = r + 1 \mod 3$,
- the edges of μ are e,
- the boldfaced edges $\bar{e}$ of μ can be chosen arbitrarily by $\forall$ subject to condition (2).

Observe that a multi-transition μ that respects (S, p, κ, r), e, and p' is unique but for the choice of the boldfaced edges $\bar{e}$.

Assume that the current position in $\mathcal{F}$ is (S, p, κ, r) and consider the following cases depending on the number of the sub-round r. In all the cases the players construct a multi-transition μ that leads to a post-state (S', p', κ', r'):

(R0) $r = 0$**:** There are two cases. If the *reach* phase is not empty i.e. $S \cap (Q^{\mathcal{B}} \times \{\mathit{reach}\}) \neq \emptyset$, then e contains all the pairs (s, s) for $s \in S$. The second case is when there are no states in the *reach* phase. We call this situation a *flush*. In that case $\forall$ can choose[2] any set $C \subseteq Q^{\mathcal{B}}$ of states q such that

$$(q, \mathit{safe}) \in S \quad \text{and} \quad \kappa(q, \mathit{safe}) > 0\ . \tag{3}$$

The chosen active states get copied to the *reach* phase thanks to the edge relation defined as: $e = \{(s, s) \mid s \in S\} \cup \big\{\big((q, \mathit{safe}), (q, \mathit{reach})\big) \mid q \in C\big\}$. In both cases the state $p' = p$ of $\mathcal{A}$ is not changed and $\forall$ chooses μ that respects (S, p, κ, r), e, and p'.

---

[2] Even if $\forall$ declares $C = \emptyset$, the fact that the *reach* phase was empty implies that we have a *flush*.

(R1) $r = 1$: $\exists$ declares: (i) a letter $a \in A$; (ii) a transition $\delta_s = (q, a, q^s_{\mathtt{L}}, q^s_{\mathtt{R}})$ of $\mathcal{B}$, for every $s = (q, z) \in S$; (iii) a transition $\delta = (p, a, p'_{\mathtt{L}}, p'_{\mathtt{R}})$ of $\mathcal{A}$. Then $\forall$ responds by selecting a direction $d \in \{\mathtt{L}, \mathtt{R}\}$. We put $p' = p'_d$, and e contains all the pairs of the form $((q, z), (q^s_d, z))$ for $s = (q, z) \in S$. $\forall$ chooses μ that respects (S, p, κ, r), e, and p'.

(R2) $r = 2$: Deterministically, every active state (q, reach) in the *reach* phase with q accepting (i.e. $q \in F$) is moved to the *safe* phase. Formally, for each $(q, z) \in S$, the relation e contains pairs $((q, z), (q, z'))$ such that either: (i) $z = z' = \mathit{safe}$; or (ii) $z = z' = \mathit{reach}$ and $q \notin F$; or (iii) $z = \mathit{reach}$, $z' = \mathit{safe}$, and $q \in F$. The state $p' = p$ of $\mathcal{A}$ is not changed. $\forall$ chooses μ that respects (S, p, κ, r), e, and p'.

If $(s, s') \in e$ we say that s' is a *μ-successor* of s. By the definition of the sub-rounds of the game, we obtain the following fact.

▶ **Fact 6.** *Every active state has between one and two μ-successors. The only case when an active state (q, z) can have two μ-successors is when $r = 0$, $z = \mathit{safe}$, and we have a* flush.

4.3 The winning condition of $\mathcal{F}$

Now we will define the winning condition for $\exists$ in $\mathcal{F}$. It will depend on the sequence of multi-transitions $\pi = \mu_0\mu_1 \dots$ that were played in $\mathcal{F}$. We will refer to the pre-state of μ_n as $(S_n, p_n, \kappa_n, r_n)$. Analogously, we will use $(S'_n, p'_n, \kappa'_n, r'_n)$ for the post-state, e_n for the edges, and $\bar{e}_n$ for the boldfaced edges of μ_n, respectively. Since π is a play, $(S'_n, p'_n, \kappa'_n, r'_n) = (S_{n+1}, p_{n+1}, \kappa_{n+1}, r_{n+1})$ and $r_n \equiv n \mod 3$.

A *trace in* π is a sequence $\alpha = s_0, s_1, \dots$ such that $(s_i, s_{i+1}) \in e_i$ of all i. A trace is *boldfaced* if $(s_i, s_{i+1}) \in \bar{e}_i$ for all i. For every $s \in S'_n$ there is a boldfaced trace ending in s, and it is unique due to condition (2); we call it the *boldfaced history of s in π*.

The winning condition will be a boolean combination of three properties of plays. We list them separately as they will be of independent interest in the proof.

W1. Infinitely many times there is a *flush* in the sub-round (R0).
W2. Some boldfaced trace changes phases infinitely many times.
W3. The sequence of states $p_0, p_1, \dots$ of the automaton $\mathcal{A}$ is accepting.

Now we declare a play to be winning for $\exists$ if it satisfies

$$\mathrm{W1} \ \wedge \ (\mathrm{W2} \vee \mathrm{W3}). \tag{4}$$

Note that Condition W2 implies Condition W1 – if some trace in a play changes phases infinitely often then the play must have infinitely many times a *flush*.

Intuitively, Condition W1 expresses that $\exists$ has not stayed forever in the *reach* phase – she has reached an accepting state of $\mathcal{B}$ whenever $\forall$ asked for it.

Condition W2 says that $\forall$ has not succeeded to bound the number of changes of phases; so he has failed to prove that on the constructed tree t he can win in $\mathcal{G}(t, K)$ for some finite K. Condition W3 takes care of the situation when the constructed tree is not in $\mathrm{L}(\mathcal{B})$. One can think of it as an escape option for $\exists$. She uses it when $\forall$ plays very cautiously and gives $\exists$ no chance to construct a trace satisfying Condition W2; an extreme example is when $\forall$ never chooses to move some active states to the *reach* phase.

▶ **Example 7.** Let $\mathcal{B}$ be the Büchi automaton from the example on page 4. Consider an automaton $\mathcal{A}$ accepting the complement of $\mathrm{L}(\mathcal{B})$, namely the set of trees with only finitely many a's on every branch. This automaton is a deterministic co-Büchi automaton having two states: p_a and p_b, with p_a being a rejecting state. So a run of $\mathcal{A}$ will be accepting if on every branch the state p_a appears only finitely often. For a state p_x the transition relation $\delta^{\mathcal{A}}$ on a letter $y \in \{a, b\}$ contains the transition (p_x, y, p_y, p_y).

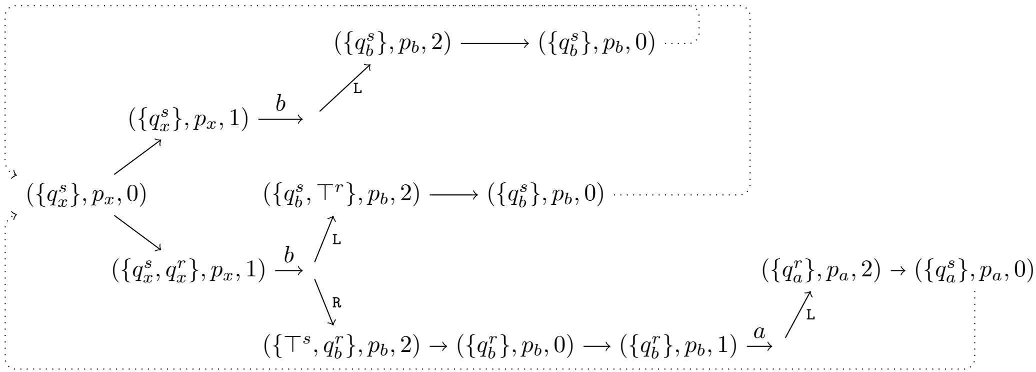

Figure 2 A symbolic representation of a strategy $\sigma_\exists$ of $\exists$. The subscript x in q^s_x or q^r_x stands for a or b. The superscript indicates if a state is in the *safe* phase or in the *reach* phase. Only branches not evidently losing for the player $\forall$ are presented.

We claim that $\exists$ has a winning strategy in $\mathcal{F}(\infty)$ constructed from $\mathcal{B}$ and $\mathcal{A}$. This strategy is schematically presented in Figure 2. For compactness of the notation we omit the third component of the position that is always ∞ and omit active states of the form $(\top, z)$ – it is trivial for $\exists$ to play from them.

The root position is a *flush*, so $\forall$ can choose whether to copy the unique active state to the *reach* phase. If he does not (i.e. he goes up in the picture) then $\exists$ chooses b and plays the transition $(q_x, b, q_b, \top)$. Then it is clear that it is better for $\forall$ to move to the left. The game gets to a similar position as in the root. If $\forall$ constantly chooses to play like this then he will lose as the play will have infinitely many times a *flush* and only states p_b, thus it will satisfy Conditions W1 and W3.

If $\forall$ decides to copy the active state during a *flush* then $\exists$ still chooses b but plays different transitions for the two copies: for the active state in the *safe* phase she chooses $(q_x, b, q_b, \top)$, for the active state in the *reach* phase she chooses $(q_x, b, \top, q_b)$. If $\forall$ chooses the left direction then the play gets to a similar position as in the root. Since there is a *flush*, and $\forall$ does not manage to see a new p_a state, the result is the same as in the case when $\forall$ has not copied the active state. If $\forall$ chooses the right direction then the play reaches a position where the only interesting active state is in the *reach* phase. Then $\exists$ chooses the letter a and the transition $(q_b, a, q_a, \top)$. It is then more interesting for $\forall$ to move to the left. The play reaches a position of the same form as the one in the root. The interesting thing that happens on this path is that the unique boldfaced trace changes phases. So, if $\forall$ chooses infinitely often to copy an active state and then go to the right then the play will satisfy Conditions W1 and W2. Otherwise there will be only finitely many occurrences of the state p_a and $\forall$ will lose since there still will be infinitely many times a *flush*.

We have already noticed that for every $K \in \overline{\omega}$, the game $\mathcal{F}(K)$ is finite. By the definition of Conditions W1, W2, and W3, the winning condition of $\mathcal{F}(K)$ is a regular property of sequences of multi-transitions. By adding multi-transitions of $\mathcal{F}(K)$ to the positions, one can obtain an equivalent game with the winning condition on sequences of positions. So up to presentation, $\mathcal{F}(K)$ is essentially a finite game with a regular winning condition, and we can solve it effectively.

▶ **Fact 8.** *For a fixed $K \in \overline{\omega}$, the winner of $\mathcal{F}(K)$ can be effectively found and he/she can win using a finite memory winning strategy. Let $m_\forall$ be the bound on the size of the memory of $\forall$ needed to win the game $\mathcal{F}(\infty)$.*

5 Characterisation

We show that $\mathcal{F}(\infty)$ characterises when $\mathrm{L}(\mathcal{B})$ is WMSO-definable. This is formulated in the following two propositions that complete the proof of Theorem 1. They rely on the following standard fact, see for instance [24].

▶ **Fact 9.** *If L is a language of a Büchi tree automaton then $L \in \mathbf{\Sigma}_1^1$. If L is a* WMSO*-definable tree language then L is Borel.*

▶ **Proposition 10.** *If $\forall$ wins $\mathcal{F}(\infty)$ then $\mathrm{L}(\mathcal{B})$ is* WMSO*-definable.*

▶ **Proposition 11.** *If $\exists$ wins $\mathcal{F}(\infty)$ then $\mathrm{L}(\mathcal{B})$ is $\mathbf{\Sigma}_1^1$-complete and not* WMSO*-definable.*

In the rest of the section we will outline the proof of Proposition 11. Suppose that $\exists$ wins in $\mathcal{F}(\infty)$. Let us fix a winning strategy $\sigma_\exists$ for $\exists$ in $\mathcal{F}(\infty)$.

We need to prove that $\mathrm{L}(\mathcal{B})$ is $\mathbf{\Sigma}_1^1$-hard, so we will construct an appropriate continuous reduction. Let $\omega\mathrm{Tr}$ denote the space of partial ω-branching trees. Such a tree τ is a non-empty, prefix-closed subset of ω^*. We say that an ω-branching tree τ is *ill-founded* if it contains an infinite branch, i.e. there exists $\alpha \in \omega^\omega$ such that for every $x \prec \alpha$ we have $x \in \tau$. We use IF to denote the set of all ill-founded ω-branching trees. If an ω-branching tree is not ill-founded then it is *well-founded*.

▶ **Fact 12.** IF *is $\mathbf{\Sigma}_1^1$-complete.*

Therefore, it is enough to construct a continuous reduction from IF to $\mathrm{L}(\mathcal{B})$. Our aim is to construct a tree $t(\tau)$ such that $t(\tau) \in \mathrm{L}(\mathcal{B})$ if and only if τ is ill-founded. The tree $t(\tau)$ will be obtained by evaluating the strategy $\sigma_\exists$ against a certain family of strategies of $\forall$, called *τ-genuine strategies*.

Let us explain this point in more detail. A strategy for $\exists$ in $\mathcal{F}(\infty)$ can be seen as a strategy tree where branching represents the choices of $\forall$ (see Figure 2). Recall from the definition of the game that $\forall$ not only chooses directions (in the sub-round (R1)), but also copies active states to the *reach* phase (during a *flush* in the sub-round (R0)), and selects boldfaced edges (in all the sub-rounds). We want to extract from the strategy tree $\sigma_\exists$ a tree where we leave only branching corresponding to the choice of directions. To this end we define τ-genuine strategies of $\forall$, where his choices to copy and to select boldfaced edges are determined by the history of the play so far. This means that a strategy tree for $\exists$ against all τ-genuine strategies of $\forall$ will be a tree with branching corresponding only to the choices of directions by $\forall$. Then we show that we have not restricted the power of $\forall$ too much, namely from this strategy tree for $\exists$ we can read out the required tree $t(\tau)$.

To properly define τ-genuine strategies of $\forall$ we will use the Kleene-Brouwer ordering, see [15, Section 2.G]. For $x, y \in \omega^*$, we say that $x \le_{\mathrm{KB}} y$ if either: (i) $x \succeq y$, or (ii) for some $n < m < \omega$ and $v, x', y' \in \omega^*$ we have $x = vnx'$ and $y = vmy'$. Intuitively, $x \le_{\mathrm{KB}} y$ if x is below or to the left of y. ϵ is the $\le_{\mathrm{KB}}$-maximal element of ω^*. There is no $\le_{\mathrm{KB}}$-minimal element in ω^*, e.g. for every $x \in \omega^*$ and its 0-successor $x \cdot 0$ we have $x \cdot 0 <_{\mathrm{KB}} x$.

▶ **Fact 13** (See [15, Proposition 2.12]). *An ω-branching tree τ is well-founded if and only if $\le_{\mathrm{KB}}$ is a well-order on the vertices of τ.*

For certain technical reasons it will be useful to have the following construction.

▶ **Definition 14.** First, assume that $\mathrm{list}\colon \omega \to \omega^*$ has the property that for each $x \in \omega^*$, the pre-image $\mathrm{list}^{-1}(\{x\})$ is infinite (i.e. every vertex appears infinitely many times in this enumeration). Now, given $x \in \omega^*$ let $\mathrm{down}(x, n)$ be either $\mathrm{list}(n)$ if $\mathrm{list}(n) <_{\mathrm{KB}} x$ or $x \cdot 0$ otherwise.

▶ **Fact 15.** *The following conditions are satisfied for every $x \in \omega^*$:*
- $\forall_{n\in\omega}$ $\mathrm{down}(x,n) <_{\mathrm{KB}} x$,
- *for every $y <_{\mathrm{KB}} x$ there are infinitely many n such that* $\mathrm{down}(x,n) = y$.

Now we can proceed with the definition of τ-genuine strategies. Our aim is to make sure that for every sequence of successive directions $d_0, d_1, \dots$ played in the sub-rounds (R1), there is a unique τ-genuine strategy of $\forall$. A τ-genuine strategy will depend on certain additional information accumulated during a play, and this information will be related to the ω-branching tree τ. Therefore, $\forall$ will keep track of an *extended position* – a position (S, p, κ, r) of $\mathcal{F}$ together with a mapping $\nu\colon S \to \tau$ and a counter $c \in \omega$. The function ν will measure the progress of every active state with respect to the $\leq_{\mathrm{KB}}$ order over τ. The counter c will count the number of times when a *flush* happened in the play.

The initial extended position of the game is the initial position of $\mathcal{F}(\infty)$ together with ν assigning to $(q^{\mathcal{B}}_{\mathrm{I}}, \mathit{safe})$ the root ϵ of τ; and the counter $c = 0$.

A strategy $\sigma_\forall$ of $\forall$ is called τ*-genuine* if it satisfies the three conditions defined below: *genuine-copying*, *flush-counting*, and *KB-tracking*.

A strategy $\sigma_\forall$ satisfies *genuine-copying* if during a *flush* in the sub-round (R0) it copies an active state s from the *safe* phase to the *reach* phase if and only if $\mathrm{down}(\nu(s), c) \in \tau$.

The condition of *flush-counting* says that $\forall$ increments c by 1 exactly when there is a *flush* in the sub-round (R0); otherwise he keeps the value c unchanged.

The last condition *KB-tracking* determines how the set of boldfaced edges $\bar{e}$ should be chosen and how to update ν. We say that a multi-transition μ from $(S, p, \kappa, r, \nu, c)$ to $(S', p', \kappa', r', \nu', c')$ with edges e and boldfaced edges $\bar{e}$ satisfies the *KB-tracking* condition when:

- If μ is not a *flush* then for every $s' \in S'$:

$$\nu'(s') = \max_{\leq_{\mathrm{KB}}} \{\nu(s) : (s, s') \in e\} \ .$$

  Moreover, the unique boldfaced edge to s' should come from s_0 realising the maximum above, i.e., $\nu'(s') = \nu(s_0)$ (if there is more than one such s_0 then we choose the smallest one according to some fixed ordering on active states).
- If μ is a *flush* then for every $s' \in S'$ from the *safe* phase, the vertex $\nu'(s')$ of τ and the boldfaced edges are determined as above. For every $s' \in S'$ in the *reach* phase there is a unique $s \in S$ with $(s, s') \in e$. This edge needs to be boldfaced and we set

$$\nu'(s') = \mathrm{down}(\nu(s), c) \ .$$

Notice that in this last case the node $\mathrm{down}(\nu(s), c)$ is in τ thanks to the *genuine-copying* condition.

▶ **Fact 16.** *Using the above notions, the following inequalities hold:*
- *if $(s, s') \in \bar{e}$ then $\nu(s) \geq_{\mathrm{KB}} \nu'(s')$,*
- *if $(s, s') \in e$ and (s, s') changes phases from safe to reach then $\nu(s) >_{\mathrm{KB}} \nu'(s')$,*
- *if $(s, s') \in e$ and (s, s') does not change phases from safe to reach then $\nu(s) \leq_{\mathrm{KB}} \nu'(s')$.*

▶ **Corollary 17.** *Suppose τ is well-founded. If π is an infinite play of $\mathcal{F}(\infty)$ consistent with a τ-genuine strategy of $\forall$ then π does not satisfy W2 (no boldfaced trace changes phases infinitely many times).*

▶ Remark. Observe that all the choices of $\forall$ except the directions d are uniquely determined in a τ-genuine strategy. Therefore, to define a τ-genuine strategy it is enough to say what will be the directions proposed by $\forall$ in the sub-rounds (R1). For the next definition it is also useful to note that all the maximal plays in $\mathcal{F}(\infty)$ are infinite.

▶ **Definition 18.** For every $\alpha \in \{\mathtt{L}, \mathtt{R}\}^\omega$ by $\sigma_\forall(\tau, \alpha)$ we denote the unique τ-genuine strategy of $\forall$ that for every $n \in \omega$ plays $d = \alpha(n)$ in the n-th sub-round (R1). Let $\pi(\tau, \alpha)$ be the infinite play of $\mathcal{F}(\infty)$ obtained when $\exists$ is playing $\sigma_\exists$ and $\forall$ is playing $\sigma_\forall(\tau, \alpha)$.

For every finite prefix $u \prec \alpha$ we denote by $\pi(\tau, u)$ the corresponding prefix of $\pi(\tau, \alpha)$. This play is defined until $\forall$ is asked to determine the $(n{+}1)$-th direction in the sub-round (R1). Let $(S_u, p_u, \kappa_u, r_u, \nu_u, c_u)$ be the extended position of this play at the beginning of the last round (i.e. when $r_u = 0$).

We can finally define the tree $t(\tau)$.

▶ **Definition 19.** We define the tree $t(\tau)$ together with a run $\rho^{\mathcal{A}}(\tau)$ of $\mathcal{A}$. For a vertex $u \in \{\mathtt{L}, \mathtt{R}\}^*$, let $t(\tau)(u)$ and $\rho^{\mathcal{A}}(\tau)(u)$ be the letter a and the state p of $\mathcal{A}$ played by $\exists$ in the sub-round (R1) of the last round of the play $\pi(\tau, u)$.

Observe that by the construction, $\rho^{\mathcal{A}}(\tau)$ is a run of $\mathcal{A}$ over $t(\tau)$. Notice also that since the strategy $\sigma_\forall(\tau, u)$ queries whether $v \in \tau$ for finitely many v at a time, the function mapping τ to $t(\tau)$ is continuous. We show that indeed the mapping is the required reduction from IF as expressed by the following two lemmas.

▶ **Lemma 20.** *If τ is well-founded then $\rho^{\mathcal{A}}(\tau)$ is accepting and thus $t(\tau) \notin \mathrm{L}(\mathcal{B})$.*

▶ **Lemma 21.** *If τ is ill-founded then $t(\tau) \in \mathrm{L}(\mathcal{B})$.*

Lemma 20 follows from Corollary 17. Consider any infinite branch α of $t(\tau)$ and the corresponding play $\pi(\tau, \alpha)$ of $\sigma_\exists$ against $\sigma_\forall(\tau, \alpha)$. Since $\sigma_\exists$ is winning we know that it satisfies the disjunction W2 $\vee$ W3. By Corollary 17 we know that no play consistent with $\sigma_\forall(\tau, \alpha)$ can satisfy Condition W2. Therefore, Condition W3 needs to be satisfied and therefore, the run $\rho^{\mathcal{A}}(\tau)$ is accepting on α.

To prove Lemma 21 we fix an ω-branching ill-founded tree $\tau \in$ IF. We then extract an accepting run $\rho^{\mathcal{B}}$ of $\mathcal{B}$ on $t(\tau)$ from the strategy $\sigma_\exists$ in $\mathcal{F}(\infty)$. The crucial point is to make sure that $\forall$ will copy infinitely often the active states of the constructed run to the *reach* phase. For this we need to rely on the condition of *genuine-copying*. Let us now describe how this construction works on our running example.

▶ **Example 22.** Recall the example from page 4 and the winning strategy for $\exists$ from the example on page 7 (see Figure 2). In this strategy $\forall$ has a choice of whether to copy or not the active state q^s_x; this corresponds to going down or up from the root, respectively. Next, $\forall$ chooses a direction. In a τ-genuine strategy a state q^s_x is assigned a node $\nu(q^s_x)$ of τ and c counts the number of flushes. The condition of *genuine-copying* requires $\forall$ to copy the state q^s_x to the *reach* phase if $\mathrm{down}(\nu(q^s_x), c) \in \tau$. Since all loops of $\sigma_\exists$ contain a *flush*, the value c counts the number of times either of the loops has been taken.

According to the definition of a τ-genuine strategy, the only moment when the value $\nu(q^s_x)$ may change is when $\forall$ copies (i.e. takes the down successor of the root at Figure 2) and in that case the value $\nu(q^r_x)$ becomes $\mathrm{down}(\nu(q^s_x), c)$ according to the *KB-tracking* condition. This becomes the new value of $\nu(q^s_x)$ if and only if the play then follows the direction $\mathtt{R}$.

If τ is well-founded then there is no infinite $\leq_{\mathrm{KB}}$-descending chain in τ and therefore, for every branch $\alpha \in \{\mathtt{L}, \mathtt{R}\}^\omega$, the play $\pi(\tau, \alpha)$ follows only finitely many times the down path of $\sigma_\exists$. Therefore, the produced tree $t(\tau)$ contains only finitely many letters a on every branch and $t(\tau) \notin \mathrm{L}(\mathcal{B})$.

Now assume that τ is ill-founded and $v_0 >_{\mathrm{KB}} v_1 >_{\mathrm{KB}} \ldots$ is an infinite $\leq_{\mathrm{KB}}$-descending chain of nodes of τ. We will use this sequence to find a branch of $t(\tau)$ that contains infinitely many a's. We start in the root of $t(\tau)$ and keep track of the current vertex $\nu(q^s_x)$ of τ. We will preserve an invariant that when being in a node $u \in \{\mathtt{L}, \mathtt{R}\}^*$ with n the number of occurrences of a on u in $t(\tau)$ then $\nu(q_x^s) = v_n$ – the vertex of τ pointed to by ν is the n-th vertex in our $\leq_{\mathrm{KB}}$-descending chain. Consider the following two cases:

1. If $\mathrm{down}(\nu(q_x^s), c) = v_{n+1}$ then $\forall$ chooses to copy, i.e., go down from the root. We follow the $\mathtt{R}$-successor in $t(\tau)$. Then $t(\tau)(u\mathtt{R}) = a$ and the game gets to the node $u\mathtt{RL}$. The number of times we have seen an a is incremented (i.e. $n' = n + 1$), and the invariant is preserved since after this loop we have $\nu(q_x^s) = v_{n+1}$.
2. Otherwise either (i) $\mathrm{down}(\nu(q_x^s), c) \notin \tau$ so $\forall$ does not copy, or (ii) $\mathrm{down}(\nu(q_x^s), c) \in \tau$ so $\forall$ copies, but we choose the direction $\mathtt{L}$. In both cases we end up in the left successor of our current node (i.e. in $u\mathtt{L}$). The new value $\nu(q_x^s)$ does not change, neither does n.

Therefore, in both cases the invariant $\nu(q_x^s) = v_n$ is preserved. Since the value of c tends to infinity, Fact 15 tells us that $\mathrm{down}(\nu(q_x^s), c) = v_{n+1}$ will eventually hold, and we will see an a. In the limit, the branch of $t(\tau)$ we follow will have infinitely many letters a.

6 Conclusions

While regular languages of infinite trees are widely used nowadays, their structure is still very poorly understood. The main reason for this is probably the lack of deterministic acceptors for such languages. This paper exhibits a gap property for languages of non-deterministic Büchi tree automata: such a language is either weakly definable, or $\mathbf{\Sigma}_1^1$-complete. Our proof uses a reduction to a finite game. Given a Büchi automaton $\mathcal{B}$, we construct a game $\mathcal{F}(\infty)$ of exponential size w.r.t. $\mathcal{B}$, and with a parity condition of size proportional to the size of $\mathcal{B}$. Thus our reduction gives an EXPTIME decision algorithm. This matches a known lower bound [25].

References

1. Mikołaj Bojańczyk. Star height via games. In *LICS*, pages 214–219, 2015.
2. Mikołaj Bojańczyk and Tomasz Idziaszek. Algebra for infinite forests with an application to the temporal logic EF. In *CONCUR*, pages 131–145, 2009.
3. Mikołaj Bojańczyk, Damian Niwiński, Alexander Rabinovich, Adam Radziwończyk-Syta, and Michał Skrzypczak. On the Borel complexity of MSO definable sets of branches. *Fundamenta Informaticae*, 98(4):337–349, 2010.
4. Mikołaj Bojańczyk and Thomas Place. Regular languages of infinite trees that are Boolean combinations of open sets. In *ICALP*, pages 104–115, 2012.
5. Julian Bradfield. The modal mu-calculus alternation hierarchy is strict. *Theoretical Computer Science*, 195:133–153, 1997.
6. Jérémie Cabessa, Jacques Duparc, Alessandro Facchini, and Filip Murlak. The Wadge hierarchy of max-regular languages. In *FSTTCS*, pages 121–132, 2009.
7. Thomas Colcombet. Fonctions régulières de coût. Habilitation thesis, Université Paris Diderot – Paris 7, 2013.
8. Thomas Colcombet, Denis Kuperberg, Christof Löding, and Michael Vanden Boom. Deciding the weak definability of Büchi definable tree languages. In *CSL*, pages 215–230, 2013.
9. Thomas Colcombet and Christof Löding. The non-deterministic Mostowski hierarchy and distance-parity automata. In *ICALP (2)*, pages 398–409, 2008.
10. Jacques Duparc, Olivier Finkel, and Jean-Pierre Ressayre. Computer science and the fine structure of Borel sets. *Theoretical Computer Science*, 257(1–2):85–105, 2001.
11. Jacques Duparc, Olivier Finkel, and Jean-Pierre Ressayre. The Wadge hierarchy of Petri nets w-languages. In *LFCS*, pages 179–193, 2013.

12 Alessandro Facchini, Filip Murlak, and Michał Skrzypczak. Rabin-Mostowski index problem: A step beyond deterministic automata. In *LICS*, pages 499–508, 2013.

13 Olivier Finkel. Borel ranks and Wadge degrees of context free omega-languages. *Mathematical Structures in Computer Science*, 16(5):813–840, 2006.

14 Tomasz Idziaszek, Michał Skrzypczak, and Mikołaj Bojańczyk. Regular languages of thin trees. *Theory of Computing Systems*, pages 1–50, 2015.

15 Alexander Kechris. *Classical descriptive set theory.* Springer-Verlag, New York, 1995.

16 Denis Kuperberg and Michael Vanden Boom. Quasi-weak cost automata: A new variant of weakness. In *FSTTCS*, volume 13 of *LIPIcs*, pages 66–77, 2011.

17 Ralf Küsters and Thomas Wilke. Deciding the first level of the μ-calculus alternation hierarchy. In *FST TCS 2002:*, volume 2556 of *LNCS*, pages 241–252, 2002.

18 Filip Murlak. The Wadge hierarchy of deterministic tree languages. *Logical Methods in Logical Methods in Comput. Sci.*, 4(4), 2008.

19 Filip Murlak. Weak index versus Borel rank. In *STACS'08*, LIPIcs, pages 573–584, 2008.

20 Damian Niwiński and Igor Walukiewicz. A gap property of deterministic tree languages. *Theor. Comput. Sci.*, 1(303):215–231, 2003.

21 Damian Niwiński and Igor Walukiewicz. Deciding nondeterministic hierarchy of deterministic tree automata. *Electronic Notes in Theoretical Computer Science*, 123:195–208, 2005.

22 Michael Oser Rabin. Weakly definable relations and special automata. In *Proceedings of the Symposium on Mathematical Logic and Foundations of Set Theory*, pages 1–23. North-Holland, 1970.

23 Jerzy Skurczyński. The Borel hierarchy is infinite in the class of regular sets of trees. *Theoretical Computer Science*, 112(2):413–418, 1993.

24 Wolfgang Thomas and Helmut Lescow. Logical specifications of infinite computations. In *REX School/Symposium*, pages 583–621, 1993.

25 Igor Walukiewicz. Deciding low levels of tree-automata hierarchy. In *Workshop on Logic, Language, Information and Computation*, volume 67 of *Electronic Notes in Theoretical Computer Science*, pages 61–75, 2002.