A Characterisation of Π^0_2 Regular Tree Languages

Filippo Cavallari[*1], **Henryk Michalewski**[2], **and Michał Skrzypczak**[†3]

1 **University of Lausanne, Department of Information Systems, Faculty of Business and Economics, Lausanne, Switzerland and University of Turin, Turin, Italy**
`filippo.cavallari@unito.it`

2 **University of Warsaw, Department of Mathematics, Informatics and Mechanics, Warsaw, Poland**
`h.michalewski@mimuw.edu.pl`

3 **University of Warsaw, Institute of Informatics, Warsaw, Poland**
`m.skrzypczak@mimuw.edu.pl`

Abstract

We show an algorithm that for a given regular tree language L decides if $L \in \mathbf{\Pi}^0_2$, that is if L belongs to the second level of Borel Hierarchy. Moreover, if $L \in \mathbf{\Pi}^0_2$, then we construct a weak alternating automaton of index $(0, 2)$ which recognises L. We also prove that for a given language L, L is definable by a weak alternating $(1, 3)$-automaton if and only if it is definable by a weak non-deterministic $(1, 3)$-automaton.

1998 ACM Subject Classification F.4.3 Formal Languages

Keywords and phrases infinite trees, Rabin-Mostowski hierarchy, regular languages

Digital Object Identifier 10.4230/LIPIcs.MFCS.2017.56

1 Introduction

Automata on infinite trees and the corresponding Monadic Second Order Logic provide a rich framework for expressing properties of regular languages of trees. Characterising natural subclasses of regular tree languages can be considered one of the fundamental problems related to automata on infinite trees. When we additionally require that the characterisation should be of an algorithmic nature, the problem usually turns to be very difficult and so far solved only in few instances.

▶ **Problem 1** (The Characterisation Problem). *For a given class of sets of trees $\mathcal{C}$ design an algorithm which decides if a given regular tree language L belongs to $\mathcal{C}$.*

Let us consider the problem for the following classes of tree languages: 1) languages definable in First Order Logic, 2) languages definable in Weak Monadic Second Order Logic, or 3) Borel languages. Providing an effective characterisation for any of the above classes among all regular languages of trees seems to be beyond the reach of currently available methods. In all the above instances it would be desirable to prove a dichotomy *simple* versus *difficult* languages; with *difficult* languages being characterised by existence of an embedding of a standard difficult language. In this work we resolve the **Characterisation Problem**

* The first author has been supported by ERC Consolidator grant LIPA (683080).
† The second and third authors have been supported by Polish National Science Centre grants 2014/13/B/ST6/03595 and 2016/22/E/ST6/00041 respectively.

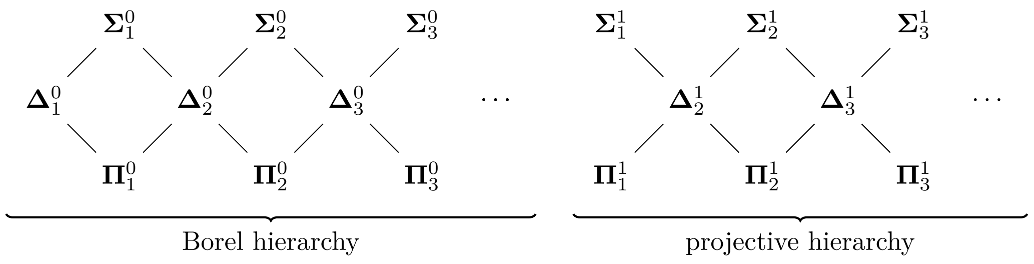

Figure 1 Borel and projective hierarchies.

for the case of $\mathcal{C} = \mathbf{\Pi}^0_2$, i.e. for the set of languages that belong to the second level of the Borel hierarchy. A posteriori, it turns out that $\mathcal{C}$ is the class of languages recognisable by weak alternating automata of *index* $(0, 2)$, that is the class of parity automata that involve priorities 0, 1, and 2; such that the transitions of the automaton are monotone wrt. priorities.

▶ **Theorem 2.** *If L is a regular tree language then either:*

- *L can be recognised by a weak alternating $(0, 2)$-automaton and so $L \in \mathbf{\Pi}^0_2$,*
- *L cannot be recognised by a weak alternating $(0, 2)$-automaton, $L \notin \mathbf{\Pi}^0_2$, and L is $\mathbf{\Sigma}^0_2$-hard.*

Moreover, it can be effectively decided which of the cases holds. If $L \in \mathbf{\Pi}^0_2$ then a weak alternating $(0, 2)$-automaton can effectively be constructed for L.

All regular languages of trees are $\mathbf{\Sigma}^1_2$ sets and from Rabin's Complementation Theorem [20] follows that every regular language of trees is in the class $\mathbf{\Delta}^1_2$. In the case of weak alternating automata (see e.g. [13]) one can provide a much more precise upper bound for the complexity:

▶ **Lemma 3** (See e.g. [6]). *If L is a language recognised by a weak alternating $(0, n)$-automaton then $L \in \mathbf{\Pi}^0_n$.*

Combining [24] and [6] we obtain that this is the optimal upper bound for the languages definable in Weak Monadic Second Order Logic.

The **Characterisation Problem** seems to be settled only for few families $\mathcal{C}$. The following list summarises all the cases which according to authors' knowledge have been considered in the literature so far, with the unrestricted input of general regular tree languages:

1. The simple case of clopen sets considered a mathematical folklore.
2. The case of open and closed sets, see [1, page 1] or [14, page 83-84]
3. The case of Boolean combinations of open sets settled in [1] using sophisticated algebraic methods.
4. The techniques of [1] were further reused in [7] to provide an effective characterisation of the class $\mathbf{\Delta}^0_2$. However, as it turned out, the application of the tools of [1] presented in that paper was not correct and the proofs contain some missing arguments[1]; as a corollary of the present article we also obtain another algorithm solving Problem 1 for the case $\mathcal{C} = \mathbf{\Delta}^0_2$. Additionally, this paper provides a new automata theoretic observation: regular languages in $\mathbf{\Delta}^0_2$ belong to the respective delta class of the weak alternating index hierarchy: for any regular language L in $\mathbf{\Delta}^0_2$ there exist a weak alternating $(0, 2)$-automaton and a weak alternating $(1, 3)$-automaton that recognise L.

[1] The statement of Theorem 1 in [7] is correct but a critical combinatorial Proposition 5 requires a different and a more sophisticated argument which will be presented in a journal version of that work.

5. From [9] we know that regular tree languages occupy the first ω–levels of Kolmogorov's $\mathcal{R}$-hierarchy.

The following problem that can be seen as a reversed version of Lemma 3 requires a fine-grained analysis of $\mathbf{\Pi}^0_n$ regular languages of trees:

▶ **Problem 4.** *Given a regular tree language that belongs to $\mathbf{\Pi}^0_n$, does there exist a weak alternating $(0, n)$-automaton that recognises it?*

In the case of $n = 1$ the positive answer is considered folklore (it also follows from [1]), however for every $n > 1$ the problem was open. Our article gives the positive answer for the specific case when $n = 2$.

Related work

Characterising WMSO among Büchi automata. The proof presented in this paper is inspired by a characterisation [23] of Borel languages (as well as recognisable by all weak alternating automata) among the languages recognisable by Büchi automata. Although the similar structure of the proof and the idea behind the characterisation game $\mathcal{F}$, there are certain differences between the two proofs. Firstly, the structure of the game $\mathcal{F}$ is much simpler here than in [23]. Moreover, the construction of the automata $\mathcal{G}_K$ from Section 6 requires certain new ideas because we deal with arbitrary parity automata (in [23] the input is restricted to Büchi automata and the construction is just an unravelling of the respective game $\mathcal{G}$). In particular, in this work we introduce the concept of K-acceptance.

Deterministic and other special classes of languages. In absence of a method solving Problem 1 for all regular languages, a number of attempts was made for special families of regular languages [18, 17, 16, 8] that are recognised by automata with restricted forms of non-determinism.

Cost-MSO and counter automata. Another take on characterising various classes of languages via games can be found in [5, 4, 3]. The authors of this paper are not aware of results directly applicable to weak alternating parity automata of index $(1, 3)$. However, as shown in this paper, weak non-deterministic parity automata of index $(1, 3)$ are equivalent with weak alternating parity automata of the same index. Therefore, it seems feasible to obtain the automata-theoretic part (without the correspondence to topological classes) of Theorem 2 using the tools presented in [5, 12, 4].

2 Basic notions

Trees. Let us fix a finite alphabet A, that is just a finite non-empty set of symbols (e.g. $A = \{a, b\}$). We work on the space of labelled complete infinite binary trees over A. An *A-labelled tree* t (shortly *tree*) is a function $t\colon \{\mathtt{L}, \mathtt{R}\}^* \to A$ where the symbols $\mathtt{L}$, $\mathtt{R}$ are called *directions*. The set of all A-labelled trees is denoted by Tr_A.

Elements $u \in \{\mathtt{L}, \mathtt{R}\}^*$ are called *nodes* of a tree. The elements $u\mathtt{L}$, $u\mathtt{R}$ are called *children* of u. The empty sequence ϵ is called the root of a tree. A *branch* of a tree is just an infinite sequence of directions $\beta \in \{\mathtt{L}, \mathtt{R}\}^\omega$. A node u is on a branch α if it is a prefix of α, i.e. $u \prec \alpha$. In that case $u = \alpha{\restriction}_{|u|}$. If u is a node of a tree then by $t{\restriction}_u$ we indicate the tree t truncated in u in the usual sense: $t{\restriction}_u(w) \overset{\text{def}}{=} t(uw)$.

Topological complexity. In this work we use the standard topological notions for the space of infinite trees, see [11, 26] . The relevant topological notions in the context of infinite trees are described in Sections 1.6.1 and 1.6.2 of [22].

The space Tr_A with the standard product topology is known to be an uncountable Polish space (homeomorphic with the Cantor set). Thus, all the standard notions of Descriptive Set Theory naturally apply to trees.

Assume that $\mathcal{X}$ is a topological space known from the context. By $\mathbf{\Sigma}^0_1$ (resp. $\mathbf{\Pi}^0_1$) we denote the set of open (resp. closed) sets in $\mathcal{X}$. The classes $\mathbf{\Sigma}^0_{n+1}$ and $\mathbf{\Pi}^0_{n+1}$ are defined inductively: $\mathbf{\Sigma}^0_{n+1}$ contains countable unions of sets in $\mathbf{\Pi}^0_n$; $\mathbf{\Pi}^0_{n+1}$ contains countable intersections of sets in $\mathbf{\Sigma}^0_n$. In particular $\mathbf{\Sigma}^0_2$ are countable unions of closed sets; this class is often denoted F_σ. Similarly, $\mathbf{\Pi}^0_2$ are countable intersections of open sets; often denoted G_δ.

For a class Γ of sets (e.g. $\mathbf{\Pi}^0_2$), we say that a set $Y \subseteq \mathcal{X}$ is Γ-*hard* if for every set $Y' \in \mathcal{X}'$ that is in Γ there exists a continuous reduction $f\colon \mathcal{X}' \to \mathcal{X}$ such that $Y' = f^{-1}[Y]$. A set Y is Γ-complete if Y is Γ-hard and belongs to Γ.

Automata Theory. In this work we use both notions of *non-deterministic* and *alternating* parity tree automata. Again, we refer the reader to [19, 25]. The notation we use comes from Sections 1.3 and 1.4 of [22].

A *parity tree automaton* $\mathcal{A}$ is a tuple $\mathcal{A} = \langle A^{\mathcal{A}}, Q^{\mathcal{A}}, q^{\mathcal{A}}_{\mathrm{I}}, \Delta^{\mathcal{A}}, \Omega^{\mathcal{A}}\rangle$, where: $A^{\mathcal{A}}$ is the alphabet we are working on; $Q^{\mathcal{A}}$ is the set of *states* of the automaton $\mathcal{A}$; $q^{\mathcal{A}}_{\mathrm{I}}$ is a particular element of $Q^{\mathcal{A}}$ and it is called the *initial state*; $\Delta^{\mathcal{A}}$ will be defined in a moment; and $\Omega^{\mathcal{A}}$ is a function $\Omega^{\mathcal{A}}\colon Q^{\mathcal{A}} \to \omega$ that assigns a *priority* to every state of the automaton. If the automaton is known from the context then we omit the superscript $\mathcal{A}$.

An automaton $\mathcal{A}$ is *non-deterministic* if $\Delta \subseteq Q \times A \times Q \times Q$ contains *transitions* of the form $(q, a, q_{\mathtt{L}}, q_{\mathtt{R}})$. A non-deterministic automaton $\mathcal{A}$ *accepts* a tree $t \in \mathrm{Tr}_A$ if there exists an *accepting run* ρ, i.e. a $Q^{\mathcal{A}}$-labelled tree that is consistent with the transitions of $\mathcal{A}$ and the parity condition is satisfied on every branch β of t: $\limsup_{n\to\infty} \Omega\big(\rho(\beta{\restriction}_n)\big)$ is even.

An automaton $\mathcal{A}$ is *alternating* if Δ is a function that assigns to each pair $q \in Q$, $a \in A$ a finite positive Boolean combination of pairs (d, q') where $d \in \{\mathtt{L}, \mathtt{R}\}$ is a direction and $q' \in Q$ is the consecutive state. For instance $\Delta(q, a)$ can be of the form $\big((\mathtt{L}, q'_{\mathtt{L}}) \wedge (\mathtt{L}, q''_{\mathtt{L}})\big) \vee (\mathtt{R}, q''_{\mathtt{R}})$.

An alternating automaton $\mathcal{A}$ induces, for every tree $t \in \mathrm{Tr}_A$, a parity game $\mathcal{A}(t)$ called the *acceptance game* of $\mathcal{A}$ on t. $\mathcal{A}$ *accepts* t if $\exists$ has a winning strategy in the game $\mathcal{A}(t)$.

We require our automata to be *complete*, meaning that for every state $q \in Q$ and letter $a \in A$ there needs to be some transition.

For both non-deterministic and alternating tree automata $\mathcal{A}$ we define the *language* of $\mathcal{A}$ (denoted $\mathrm{L}(\mathcal{A})$) as the set of all trees accepted by $\mathcal{A}$. It is known that the expressive power of non-deterministic and alternating automata is the same:

▶ **Theorem 5** ([10]). *Let $L \subseteq \mathrm{Tr}_A$. There exists an alternating parity tree automaton $\mathcal{A}$ such that $\mathrm{L}(\mathcal{A}) = L$ if and only if there exists a non-deterministic parity tree automaton $\mathcal{B}$ such that $\mathrm{L}(\mathcal{B}) = L$. Moreover, both translations are effective.*

If for $L \subseteq \mathrm{Tr}_A$ there exists a non-deterministic (equivalently alternating) automaton $\mathcal{A}$ such that $L = \mathrm{L}(\mathcal{A})$ then we say that L is *regular*. A parity automaton is *weak* if the values of Ω are non-decreasing along transitions. The *index* of an automaton is the pair (i, j) where i is the minimal and j is the maximal value of Ω on Q.

3 Overview of the proof of Theorem 2

Let L be a regular language and let us fix once and for all two non-deterministic parity tree automata $\mathcal{A}$ and $\mathcal{B}$ that recognise respectively: $\mathrm{L}(\mathcal{B}) = L$ is the given language and $\mathrm{L}(\mathcal{A}) = L^{\mathrm{c}}$ is its complement. The proof will consist of the following steps:

- First we define a game $\mathcal{F}$ of infinite duration and perfect information. The game $\mathcal{F}$ is played by two players: Eve ($\exists$) and Adam ($\forall$). Player $\exists$ constructs a tree t together with three runs: one of the automaton $\mathcal{A}$ and two of the automaton $\mathcal{B}$. The second of them is influenced by $\forall$ who can ask $\exists$ to *restart* whenever he wants. The crucial property of the game $\mathcal{F}$ is that it is played over a finite arena and the winning condition is ω-regular.
- If $\exists$ wins $\mathcal{F}$ then her winning strategy can be used to prove that the language $\mathrm{L}(\mathcal{B})$ is actually $\boldsymbol{\Sigma}_2^0$-hard. In particular it cannot be recognised by a weak alternating automaton of index $(0, 2)$. To prove this topological hardness we test the winning strategy of $\exists$ against a well-designed family of strategies of $\forall$. In terms of Descriptive Set Theory it can be seen as finding an embedding of the Cantor set 2^ω that intersects $\mathrm{L}(\mathcal{B})$ on rationals.
- If $\forall$ wins $\mathcal{F}$ then we use his finite-memory winning strategy to construct a finite approximation of the automaton $\mathcal{B}$ that is denoted $\mathcal{G}_{K_0}$. The construction ensures that $\mathcal{G}_{K_0}$ is a weak alternating automaton of index $(0, 2)$ that recognises $\mathrm{L}(\mathcal{B})$.

4 The game $\mathcal{F}$

We start by defining a game $\mathcal{F}$ of infinite duration that is based on the non-deterministic parity tree automata $\mathcal{A} = \langle A, Q^{\mathcal{A}}, q_{\mathrm{I}}^{\mathcal{A}}, \Delta^{\mathcal{A}}, \Omega^{\mathcal{A}}\rangle$ and $\mathcal{B} = \langle A, Q^{\mathcal{B}}, q_{\mathrm{I}}^{\mathcal{B}}, \Delta^{\mathcal{B}}, \Omega^{\mathcal{B}}\rangle$ for L^{c} and L respectively. The purpose of $\mathcal{F}$ is to satisfy the following two propositions.

▶ **Proposition 6.** *If $\exists$ wins $\mathcal{F}$ then $\mathrm{L}(\mathcal{B})$ is $\boldsymbol{\Sigma}_2^0$-hard.*

▶ **Proposition 7.** *If $\forall$ wins $\mathcal{F}$ then $\mathrm{L}(\mathcal{B})$ is recognised by a weak alternating $(0, 2)$-automaton.*

The above propositions together with Lemma 3 give a complete characterisation of the topological complexity and the weak index of $\mathrm{L}(\mathcal{B})$.

Positions of $\mathcal{F}$. The positions of $\mathcal{F}$ are of the form $(p, q, s) \in Q^{\mathcal{A}} \times Q^{\mathcal{B}} \times Q^{\mathcal{B}}$ where: $p \in Q^{\mathcal{A}}$ is called an *$\mathcal{A}$-state*, $q \in Q^{\mathcal{B}}$ is called a *$\mathcal{B}$-state*, $s \in Q^{\mathcal{B}}$ is called an *active state*. The initial position of $\mathcal{F}$ is $(q_{\mathrm{I}}^{\mathcal{A}}, q_{\mathrm{I}}^{\mathcal{B}}, q_{\mathrm{I}}^{\mathcal{B}})$.

Rounds of $\mathcal{F}$. Assume that a round of $\mathcal{F}$ starts in a position (p, q, s). The choices done by the players are as follows:

1. $\forall$ can choose to *restart* by letting $s' = q$ or to *stay* by keeping $s' = s$.
2. $\exists$ declares: (i) a letter $a \in A$; (ii) a transition $(p, a, p_{\mathtt{L}}, p_{\mathtt{R}}) \in \Delta^{\mathcal{A}}$ of $\mathcal{A}$; (iii) a transition $(q, a, q_{\mathtt{L}}, q_{\mathtt{R}}) \in \Delta^{\mathcal{B}}$ of $\mathcal{B}$; (iv) another transition $(s', a, s'_{\mathtt{L}}, s'_{\mathtt{R}}) \in \Delta^{\mathcal{B}}$ of $\mathcal{B}$.
3. $\forall$ responds by selecting a direction $d \in \{\mathtt{L}, \mathtt{R}\}$.

After such a round the game proceeds to the position (p_d, q_d, s'_d). Four example rounds of $\mathcal{F}$ are presented in Figure 2.

If π is a finite or infinite play of $\mathcal{F}$, a *trace* is a finite or infinite sequence of active states s in consecutive rounds in which $\forall$ has not *restarted*. Thus, those active states come from successive transitions of the automaton $\mathcal{B}$.

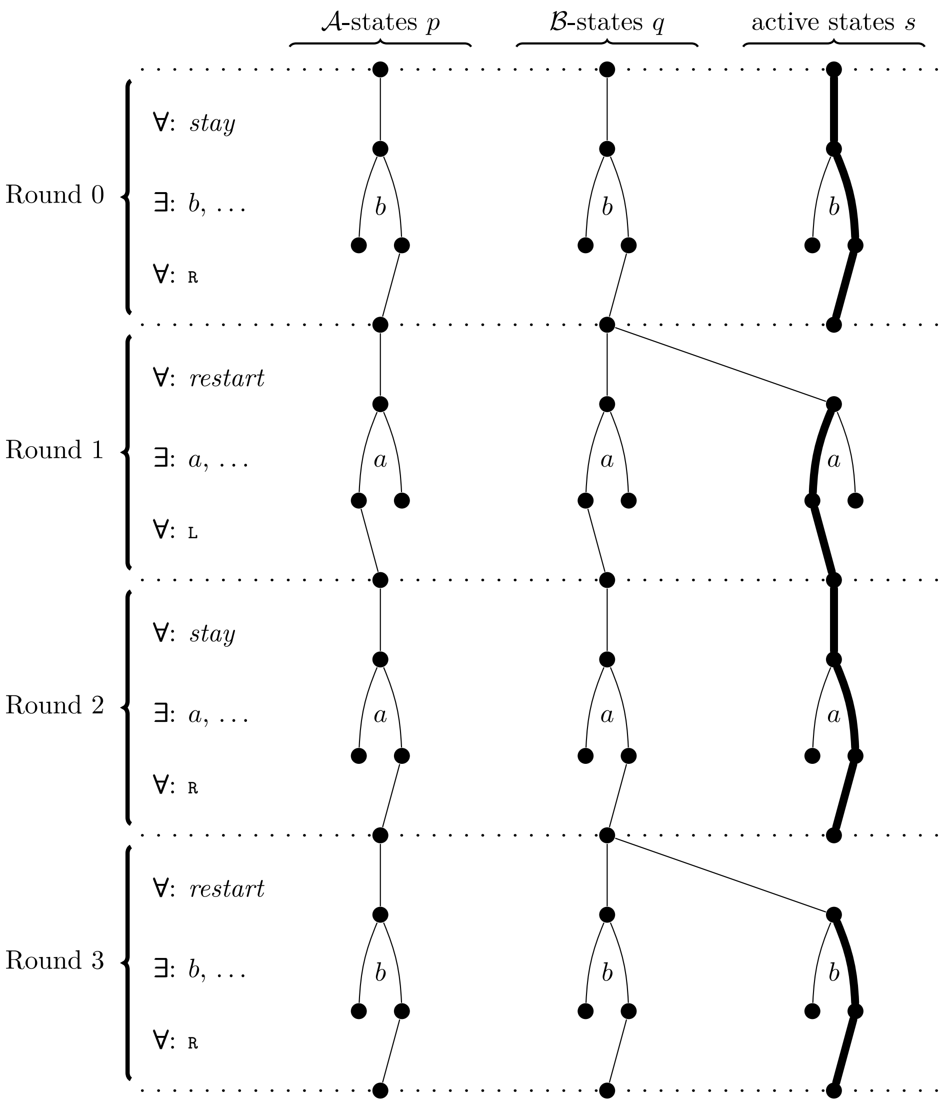

Figure 2 Four consecutive rounds of the game $\mathcal{F}$. The black dots are the states of the automata $\mathcal{A}$ and $\mathcal{B}$. Each round consists of three choices: first ∀ either *restarts* or *stays*, then ∃ provides a letter and three transitions (depicted by those Λ-shaped gadgets), finally ∀ chooses a direction. The three boldfaced paths are three traces formed by the active states: the first one lasts in Round 0; the second one in Rounds 1 and 2; the third one starts in Round 3.

Winning condition of $\mathcal{F}$. Now we will define the winning condition for ∃ in $\mathcal{F}$. It will depend on a Boolean combination of the following three properties, speaking about the sequence of rounds that were played:

(WR) ∀ has *restarted* infinitely many times.

(WA) The sequence of $\mathcal{A}$-states p is accepting in $\mathcal{A}$.

(WB) The sequence of active states s is accepting in $\mathcal{B}$ (i.e. it satisfies the parity condition).

A play of $\mathcal{F}$ is winning for ∃ if it satisfies

$$\big((\mathrm{WR}) \wedge (\mathrm{WA})\big) \ \vee \ \big(\neg(\mathrm{WR}) \wedge (\mathrm{WB})\big). \tag{1}$$

In other words, there are two cases: If $\forall$ has restarted infinitely many times then $\exists$ wins iff the sequence of visited $\mathcal{A}$-states satisfies the parity condition. If $\forall$ has restarted only finitely many times then $\exists$ wins iff the sequence of visited active states satisfies the parity condition. Notice that a priori both (WA) and (WB) can happen simultaneously, because for example there may exist two runs $\rho_{\mathcal{A}}$ and $\rho_{\mathcal{B}}$ of the automata $\mathcal{A}$ and $\mathcal{B}$ that both satisfy the parity condition on some particular branch.

By the definition, the winning condition of $\mathcal{F}$ is an ω-regular property of sequences of rounds. Additionally, there are only finitely many positions of $\mathcal{F}$ and each round allows finitely many possible choices by the players. Therefore, we obtain the following fact.

▶ **Fact 8** ([2]). *The winner of $\mathcal{F}$ can be effectively found and he/she can win using a finite memory winning strategy.*

5 Proof of Proposition 6

In this section we prove that if $\exists$ wins $\mathcal{F}$ then $\mathrm{L}(\mathcal{B})$ is $\mathbf{\Sigma}^0_2$-hard. Let $\sigma_\exists$ be her winning strategy. Let $C \subseteq \{0,1\}^\omega$ be the set of sequences containing only finitely many 1s.

It is known that C is $\mathbf{\Sigma}^0_2$-complete [26]. We will construct a continuous reduction from C to $\mathrm{L}(\mathcal{B})$ and so we obtain that $\mathrm{L}(\mathcal{B})$ is $\mathbf{\Sigma}^0_2$-hard.

We will say that σ is a *quasi-strategy* of $\forall$ in $\mathcal{F}$ if σ specifies when to *restart* and leaves undecided the choice of directions d. Notice that if σ is a quasi-strategy of $\forall$ then we can construct a tree t consisting of the letters a played by $\sigma_\exists$ against σ: the letter $t(u)$ is the ($|u|+1$)th letter played by $\sigma_\exists$ against $\forall$ playing accordingly to σ and choosing successive directions of u.

To each sequence $\alpha \in \{0,1\}^\omega$ we will assign a quasi-strategy σ_α of $\forall$ in $\mathcal{F}$. Consider $\alpha \in \{0,1\}^\omega$ and an Mth round of $\mathcal{F}$ for $M = 0, 1, \ldots$

- If $\alpha(M) = 0$ then σ_α *stays* by keeping $s' = s$.
- If $\alpha(M) = 1$ then σ_α *restarts* by putting $s' = q$.

Let the tree t_α be the effect of confronting the strategy $\sigma_\exists$ against the quasi-strategy σ_α. Since the behaviour of the strategy σ_α in an Mth round of $\mathcal{F}$ depends only on the first M bits of α, the function $\alpha \mapsto t_\alpha$ is continuous. A routine verification (see below) shows that

$$\alpha \in C \iff t_\alpha \in \mathrm{L}(\mathcal{B}). \tag{2}$$

When $\alpha \in C$. First assume that $\alpha \in C$, i.e. that there are only finitely many 1s in α. Let M be the maximal number such that $\alpha(M-1) = 1$ (or $M = 0$ if there is no such M). Let $\rho^{\mathcal{B}}$ be the run of $\mathcal{B}$ defined as follows:

- For $|u| \leq M$ let $\rho^{\mathcal{B}}(u)$ be the $\mathcal{B}$-state q from the beginning of the $|u|$th round of the play consistent with $\sigma_\exists$ and σ_α in which the sequence of directions chosen by $\forall$ was u.
- For $|u| > M$ let $\rho^{\mathcal{B}}(u)$ be the active state s from the beginning of the $|u|$th round of the play consistent with $\sigma_\exists$ and σ_α in which the sequence of directions chosen by $\forall$ was u.

It is easy to see that $\rho^{\mathcal{B}}$ is in fact a run of $\mathcal{B}$ over t_α. It remains to see that it is accepting. Consider an infinite branch of t_α. This branch corresponds to an infinite play of $\mathcal{F}$ consistent with $\sigma_\exists$ and the quasi-strategy σ_α. Since in all the rounds after the Mth one, $\forall$ has *stayed* by putting $s' = s$, the states of $\rho^{\mathcal{B}}$ form an infinite trace in that play. Therefore, the condition $\neg$(WR) holds. As the play is won by $\exists$, also (WB) must hold. It means that the trace must be accepting in $\mathcal{B}$, thus the run $\rho^{\mathcal{B}}$ is accepting on our branch. This way we have proved that $\rho^{\mathcal{B}}$ is accepting and $t_\alpha \in \mathrm{L}(\mathcal{B})$.

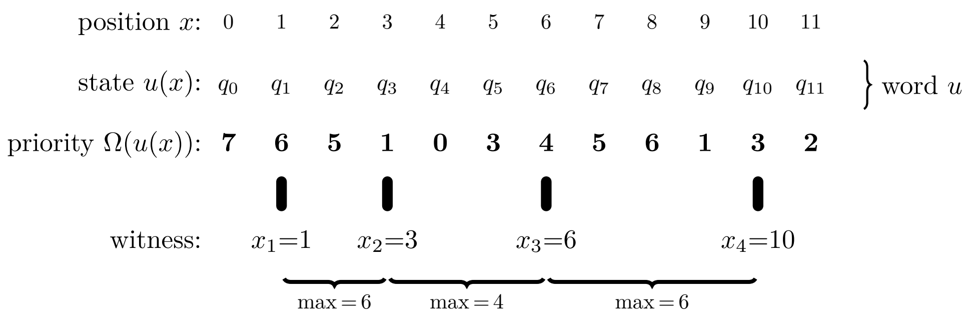
position x: 0 1 2 3 4 5 6 7 8 9 10 11
state $u(x)$: q_0 q_1 q_2 q_3 q_4 q_5 q_6 q_7 q_8 q_9 q_{10} q_{11}
word u
priority $\Omega(u(x))$: 7 6 5 1 0 3 4 5 6 1 3 2
witness: x_1=1 x_2=3 x_3=6 x_4=10
max = 6
max = 4
max = 6

Figure 3 A word u that is 4-accepting. The sequence x_1, x_2, x_3, x_4 witnesses that. If we loop u between the positions 1 and 6, we get the sequence $q_0, q_1, \ldots, q_6, q_1, q_2, \ldots, q_6, q_1 \ldots$ that satisfies the parity condition.

When $\alpha \notin C$. Now assume that $\alpha \notin C$, i.e. that there are infinitely many 1s in α. Our aim is to prove that the run $\rho^{\mathcal{A}}$ formed by the $\mathcal{A}$-states p played by $\exists$ in all the plays consistent with $\sigma_\exists$ and σ_α is accepting. Consider an infinite branch of t_α and the corresponding play π of $\mathcal{F}$. Since infinitely many times $\forall$ has *restarted*, this play satisfies (WR). As the play is won by $\exists$, also (WA) must hold. It means that the sequence of $\mathcal{A}$-states p must be accepting in $\mathcal{A}$. Thus, we have proved that $t_\alpha \in \mathrm{L}(\mathcal{A})$ and therefore $t_\alpha \notin \mathrm{L}(\mathcal{B})$. This concludes the proof of $\mathbf{\Sigma}^0_2$-hardness of $\mathrm{L}(\mathcal{B})$.

6 Proof of Proposition 7

In this section we prove that if $\forall$ wins $\mathcal{F}$ then $\mathrm{L}(\mathcal{B})$ can be recognised by a weak alternating parity automaton of index $(0, 2)$. Since the winning condition of $\mathcal{F}$ is ω-regular, we can assume that $\forall$ wins using a strategy $\sigma_\forall$ based on a finite-memory structure M. Our aim is to construct an automaton recognising $\mathrm{L}(\mathcal{B})$.

***K*-accepting runs.** We start by defining a notion of K-accepting sequences — sequences of states of $\mathcal{B}$ that are similar to accepting ones. We will show that the strategy $\sigma_\forall$ must avoid such sequences.

Let u be a finite or infinite sequence of states of $\mathcal{B}$. Consider a number $K \in \omega$. We say that u is K*-accepting* if there exists a sequence of positions $0 \le x_1 < x_2 < \ldots < x_K < |u|$ such that for every $n = 1, 2, \ldots, K-1$ we have:

$$\max \left\{ \Omega^{\mathcal{B}}\big(u(x)\big) \;\middle|\; x_n \le x \le x_{n+1} \right\} \text{ is even.} \tag{3}$$

In other words, for $n = 1, \ldots, K-1$, the maximal priority of states between positions x_n and x_{n+1} of u must be even. We call such a sequence of positions $(x_1, \ldots, x_K)$ a *witness* of K-acceptance, see Figure 3.

The above definition is constructed in such a way to guarantee the following properties:

(P1) If u is K-accepting then it contains K positions such that each cycle built using an interval between two of them gives us a sequence of states satisfying the parity condition.

(P2) For every K, the set of all finite words that are K-accepting is regular. It is not obvious how a regular expression for this language should look like, however, the definition of the property of being K-accepting is clearly MSO-definable, thus by the results of Rabin, Scott [21], and Trakhtenbrot [27] (cf. e.g. [19]), we know that this language is regular.

(P3) If u is K-accepting then every word of the form uw is also K-accepting.

(P4) If $\alpha \in \left(Q^{\mathcal{B}}\right)^{\omega}$ satisfies the parity condition then for every $K \in \omega$ there exists a finite prefix of α that is K-accepting.

▶ **Lemma 9.** *There exists a value $K_0 \in \omega$ such that if π is an infinite play of $\mathcal{F}$ consistent with $\sigma_{\forall}$ then no trace of π is K_0-accepting.*

Proof. Let $K_0 \stackrel{\text{def}}{=} \left(|Q^{\mathcal{A}}| \times |Q^{\mathcal{B}}| \times |Q^{\mathcal{B}}|\right) \times |M| + 1$ where inside the brackets is the number of positions of $\mathcal{F}$ and M is the memory structure of $\sigma_{\forall}$.

Assume for the sake of contradiction that there exists a play π that is consistent with $\sigma_{\forall}$ and contains a K_0-accepting trace. For a round number $x \in \omega$ during π let (v_x, m_x) be the configuration of the game at the moment when x rounds were played: $v_x = (p_x, q_x, s_x)$ for an $\mathcal{A}$-state p_x, $\mathcal{B}$-state q_x, and active state s_x; and m_x is the current memory value of $\sigma_{\forall}$.

By the assumption we know that for some $x < y < \omega$ the sequence of active states $s_x,\ s_{x+1},\ \ldots,\ s_y$ is a trace (i.e. there is no *restart* during these rounds) and it is K_0-accepting. Let $x \leq x_1, \ldots, x_{K_0} \leq y$ be a sequence of numbers of rounds that is a witness for the K_0-acceptance of this trace, see Equation (3).

Figure 4 provides an illustration for this construction. The upper picture presents a play π seen in the product of the game $\mathcal{F}$ and the memory structure M used by $\sigma_{\forall}$. Small dots mark positions before successive rounds of this play. The lower picture presents the play π in a chronological way. The boldfaced vertical snake-like shape is a trace that is 5-accepting; the rounded shapes indicate a witness of this fact: $x_1 = 2$, $x_2 = 3$, $x_3 = 6$, $x_4 = 8$, and $x_5 = 10$. Since 5 is bigger than the number of available pairs (v, m) we have a repetition: $(v_3, m_3) = (v_8, m_8)$. This allows us to construct a new play π', by staying forever on the loop between the rounds 3 and 8. The play π' obtained this way contains an infinite trace that satisfies the parity condition.

By the choice of K_0 we know that for some $1 \leq n < n' \leq K_0$ we have: $v_{x_n} = v_{x_{n'}}$ and $m_{x_n} = m_{x_{n'}}$; i.e. there must be a repetition of the position of $\mathcal{F}$ and the memory of $\sigma_{\forall}$ among the positions witnessing K_0-acceptance of the trace.

Consider a play π' of $\mathcal{F}$ which starts as π for the first x_n rounds. Then π' follows the loop between the rounds $x_n + 1$ and $x_{n'}$. Notice that π' is in fact a play because $v_{x_n} = v_{x_{n'}}$. Since we have chosen the positions x_n, $x_{n'}$ from a trace, this loop does not contain a *restart*. Clearly π' is consistent with $\sigma_{\forall}$ because the memory values m_{x_n} and $m_{x_{n'}}$ are equal.

Because x_n and $x_{n'}$ are chosen from a witness of K_0-acceptance of the trace, Property (P1) implies that π' contains an infinite accepting trace. Therefore, the play π' satisfies $\big(\neg(\text{WR}) \wedge (\text{WB})\big)$ and thus is winning for $\exists$ in $\mathcal{F}$, what contradicts the assumption that $\sigma_{\forall}$ was a winning strategy of $\forall$. ◀

Construction of automata $\mathcal{G}_K$. Take a number $K \in \omega$. We will now define a weak alternating parity automaton $\mathcal{G}_K$ of index $(0, 2)$. The language $\mathrm{L}(\mathcal{G}_K)$ will be an over-approximation of $\mathrm{L}(\mathcal{B})$. Later on we will prove that the strategy $\sigma_{\forall}$ witnesses the fact that $\mathrm{L}(\mathcal{B})$ actually equals $\mathrm{L}(\mathcal{G}_K)$ for some $K \in \omega$ (in fact for $K = K_0$ from Lemma 9).

The idea behind the automaton $\mathcal{G}_K$ is the following: $\mathcal{G}_K$ accepts a tree t if there exists a run $\rho_0^{\mathcal{B}}$ of $\mathcal{B}$ over t such that for every node u of the tree, it is possible to find another run of $\mathcal{B}$ over the subtree $t{\restriction}_u$ starting from the state $\rho_0^{\mathcal{B}}(u)$ that is K-accepting on every branch of this subtree.

Assume that $\mathcal{D}(K) = \langle Q^{\mathcal{B}}, Q^{\mathcal{D}}, q_{\mathrm{I}}^{\mathcal{D}}, \Delta^{\mathcal{D}}, F^{\mathcal{D}} \rangle$ is a deterministic automaton over finite words (DFA) with the alphabet $Q^{\mathcal{B}}$ that recognises the language of K-accepting sequences of states of $\mathcal{B}$, see Property (P2).

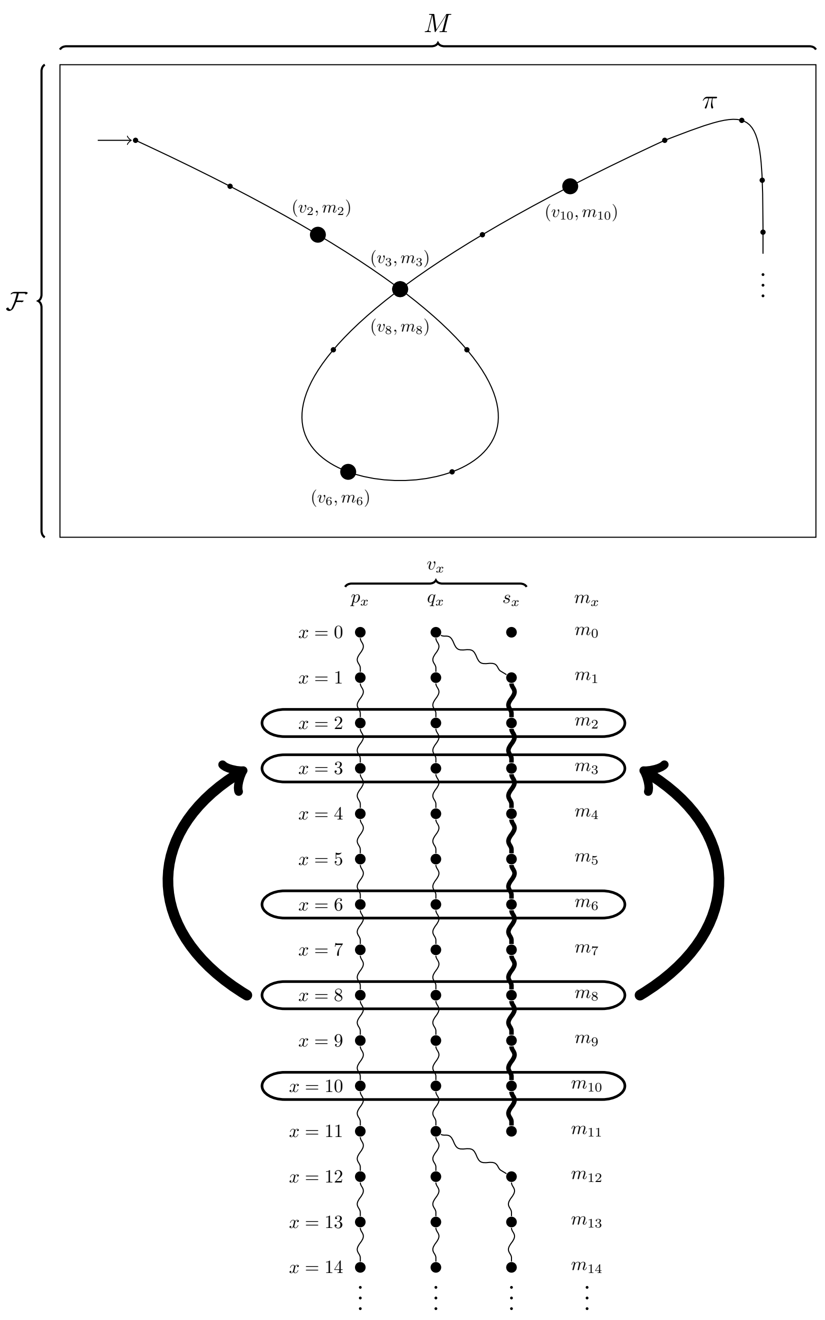

Figure 4 An illustration to the proof of Lemma 9.

The states of $\mathcal{G}_K$ are of the form (q, τ) where $q \in Q^{\mathcal{B}}$ is a state of $\mathcal{B}$ and $\tau \in \{\mathbf{?}\} \sqcup Q^{\mathcal{D}}$ is either $\mathbf{?}$ or a state of $\mathcal{D}(K)$.

The initial state of $\mathcal{G}_K$ is $(q^{\mathcal{B}}_{\mathrm{I}}, \mathbf{?})$. The transitions of $\mathcal{G}_K$ are built by the following rules. Given a state (q, τ) and a letter a, the successive state and direction are constructed in the following way (formally the following choices should be encoded as a finite positive Boolean combination of the consecutive directions and states).

1. If $\tau = \mathbf{?}$ then $\forall$ can choose to *start* by letting $\tau' = q^{\mathcal{D}}_{\mathrm{I}}$ or to *skip* by keeping $\tau' = \mathbf{?}$. If $\tau \in Q^{\mathcal{D}}$ then $\forall$ has no choice and in that case $\tau' = \tau$.

2. If $\tau' \in Q^{\mathcal{D}}$ then we let[2] $\tau'' = \Delta^{\mathcal{D}}(\tau, q)$, otherwise $\tau'' = \tau' = \mathbf{?}$ is unchanged.
3. $\exists$ proposes a transition of $\mathcal{B}$ of the form $\big(q, a, q_{\mathtt{L}}, q_{\mathtt{R}}\big)$.
4. $\forall$ chooses a direction $d \in \{\mathtt{L}, \mathtt{R}\}$.

After these choices are done, the automaton moves in the direction d to the state (q_d, τ''). Let the priority of a state (q, τ) of $\mathcal{G}_K$ be: (i) If $\tau = \mathbf{?}$ then the priority is 0. (ii) If $\tau \in Q^{\mathcal{D}} \setminus F^{\mathcal{D}}$ then the priority is 1. (iii) If $\tau \in F^{\mathcal{D}}$ then the priority is 2.

Notice that because of the structure of the transitions of $\mathcal{G}_K$, the above defined condition is a weak parity condition of index $(0, 2)$. It is important to notice that Property (P3) implies that once a state of priority 2 is reached then we never move to a state of priority 1.

▶ **Lemma 10.** *For every $K \in \omega$ we have* $\mathrm{L}(\mathcal{B}) \subseteq \mathrm{L}(\mathcal{G}_K)$.

Proof. Take a tree $t \in \mathrm{L}(\mathcal{B})$ and let $\rho^{\mathcal{B}}$ be an accepting run of $\mathcal{B}$ on t. Then clearly $\exists$ can win the acceptance game $\mathcal{G}_K(t)$ by just playing consecutive transitions of $\rho^{\mathcal{B}}$. When $\forall$ chooses at some point to *start*, ultimately a state with $\tau \in F^{\mathcal{D}}$ will be reached because of Property (P4). Therefore, every play will be won by $\exists$. ◀

Equivalence. We will now conclude the proof of Proposition 7 using the following lemma.

▶ **Lemma 11.** *For K_0 from Lemma 9 we have* $\mathrm{L}(\mathcal{G}_{K_0}) = \mathrm{L}(\mathcal{B})$.

Proof. Assume contrarily that $\mathrm{L}(\mathcal{G}_{K_0}) \neq \mathrm{L}(\mathcal{B})$. Lemma 10 says that $\mathrm{L}(\mathcal{B}) \subseteq \mathrm{L}(\mathcal{G}_{K_0})$, so there must exists a tree $t \in \mathrm{L}(\mathcal{G}_{K_0}) \setminus \mathrm{L}(\mathcal{B})$. From that assumption we know that:

- there exists an accepting run $\rho^{\mathcal{A}}$ of $\mathcal{A}$ over t,
- $\exists$ has a winning strategy $\delta_\exists$ in the acceptance game $\mathcal{G}_{K_0}(t)$.

Our aim is to prove that $\exists$ can win in $\mathcal{F}$ against $\sigma_\forall$ by using a strategy $\sigma_\exists$ that is based on $\rho^{\mathcal{A}}$ and $\delta_\exists$. Let us define the strategy $\sigma_\exists$.

First, $\sigma_\exists$ plays the letters a and the transitions of $\mathcal{A}$ from the $\mathcal{A}$-states p according to the tree t and the run $\rho^{\mathcal{A}}$. This way we guarantee that every play of this strategy will satisfy (WA). Additionally $\sigma_\exists$ chooses the transitions of $\mathcal{B}$ from the $\mathcal{B}$-states q according to the strategy $\delta_\exists$ simulating the situation that $\forall$ has never *started*. Thus, at every moment of a play consistent with $\sigma_\exists$, there is a unique play of $\delta_\exists$ ending in a node u and a state of $\mathcal{G}_{K_0}$ of the form $(q, \mathbf{?})$ with u being the sequence of directions played so-far by $\forall$ in $\mathcal{F}$ and q being the current $\mathcal{B}$-state. What remains is the choice of transitions of $\mathcal{B}$ from the active states s. For that, $\exists$ will keep track of a play of the acceptance game $\mathcal{G}_{K_0}(t)$ with $\tau \in Q^{\mathcal{D}}$. At the initial position of $\mathcal{F}$ the play is the one which begins by $\forall$ *starting* (i.e. $\tau = q_{\mathrm{I}}^{\mathcal{D}}$). Whenever $\forall$ *restarts* in $\mathcal{F}$, $\exists$ forgets about the previously tracked play of $\mathcal{G}_{K_0}(t)$ and begins to track the play that comes with the current $\mathcal{B}$-state q by simulating the situation in which $\forall$ has just *started* in $\mathcal{G}_{K_0}(t)$.

Consider the play π that is consistent with both $\sigma_\exists$ and $\sigma_\forall$. We need to prove that π is winning for $\exists$. As we have already observed, such a play satisfies (WA). We will prove that it also satisfies (WR) by proving the following claim. This concludes the proof of Lemma 11 by giving a contradiction: $\sigma_\forall$ is a winning strategy of $\forall$ but π is a play consistent with $\sigma_\forall$ that is winning for $\exists$.

▶ **Claim 12.** *In the play π Player $\forall$ must have* restarted *infinitely many times.*

[2] We follow the transition of $\mathcal{D}$ from τ over q: $\tau \in Q^{\mathcal{D}}$ is a state of $\mathcal{D}$ and $q \in Q^{\mathcal{B}}$ is a letter read by $\mathcal{D}$.

Assume contrarily that from some point on ∀ has not *restarted*. Thus, π contains an infinite trace on which ∃ has played successive transitions of $\mathcal{B}$ in the active states s according to her strategy $\delta_\exists$ in $\mathcal{G}_{K_0}(t)$. Since the strategy $\delta_\exists$ is winning, some prefix of the considered trace must be K_0-accepting. This gives a contradiction with Lemma 9. ◀

7 Weak non-deterministic $(1, 3)$-automata

In this section we prove the following additional result that may be considered folklore, although we have not found it in the literature. The construction is based on the standard de-alternation techniques together with the idea from [15].

▶ **Theorem 13.** *If L is a language that can be recognised by a weak alternating parity automaton $\mathcal{A}$ of index $(1,3)$ then L can be recognised by a weak non-deterministic parity automaton $\mathcal{B}$ of index $(1,3)$.*

This observation was important to properly define the game $\mathcal{F}$. However, somehow surprisingly, it does not play any role in the final proof of Theorem 2.

The idea of the proof is as follows. Given a tree $t \in \mathrm{Tr}_A$ the automaton $\mathcal{B}$ will guess a positional strategy of ∃ in the acceptance game $\mathcal{A}(t)$. Then it will verify that the guessed strategy is in fact winning. Therefore, it will track all the possible choices performed by ∀ along all the branches of the tree t. Thus, the set of states of $\mathcal{B}$ is the power set $\mathsf{P}(Q^{\mathcal{A}})$. Notice that the guessed strategy of ∃ is winning if for every branch of t the following conditions are satisfied: (i) no state of $\mathcal{A}$ of priority 3 is ever reached, (ii) every play ultimately reaches a state of priority 2.

An easy application of König's lemma shows that the second condition above actually implies that at some point no state of priority 1 can belong to the set of reachable states of $\mathcal{A}$. This way, the automaton $\mathcal{B}$ can be seen as a naïve power set construction over $\mathcal{A}$. Such a construction can be performed for any alternating automaton (even not weak), however, in most of the cases the assignment of priorities to the states of the power set automaton is not correct. The crucial ingredient of this construction relies on the fact that the weak parity condition of index $(1,3)$ admits a correct priority assignment for the power set automaton.

8 Conclusions and further work

This work provides a relatively simple effective characterisation of the class of regular languages in $\mathbf{\Pi}_2^0$. Additionally, it proves that the considered class of languages coincides with the respective level of the alternating index hierarchy (i.e. weak alternating $(0,2)$-automata).

The simplicity of involved techniques comes from certain specific properties of the considered classes. Firstly, there are ω-regular languages that are complete for the class $\mathbf{\Pi}_2^0$. In our case the examples are: the language C of infinite binary sequences containing infinitely many 1s; and the property (WR) used in the winning condition of the game $\mathcal{F}$. Secondly, similarly to the case of Büchi languages, the class of weak alternating $(1,3)$-automata admits a dealternation technique, see Theorem 13. Although this dealternation result does not play any role in the proof of Theorem 2, it was used in the design of the game $\mathcal{F}$ and stays behind the fact that the game actually characterises the class of languages recognisable by weak alternating $(0,2)$-automata.

We plan to investigate generalisations of Theorem 2 to $\mathbf{\Pi}_n^0$ for $n \geq 3$. Since decidability results related to the **Characterisation Problem** are not that easy to obtain, we propose for further investigation the topological problems related to the **Characterisation Problem** and Problem 4 formulated in Introduction.

Moreover, our work gives a quite clear situation up to the second level of Borel Hierarchy in terms of decidability and correspondence between the Borel index and the weak index, but there are still some "holes" that have to be filled regarding e.g. *Wadge Hierarchy*:

▶ **Problem 14.** *Find all Wadge degrees inhabited by regular $\mathbf{\Delta}^0_2$ languages of trees.*

From [6] it follows that every Wadge degree less than ω^ω is inhabited by a regular language. We believe that this is the maximum regular languages can get:

▶ **Conjecture 15.** *There is no regular tree language in $\mathbf{\Delta}^0_2$ with Wadge degree above ω^ω.*

References

1 Mikołaj Bojańczyk and Thomas Place. Regular languages of infinite trees that are Boolean combinations of open sets. In *ICALP*, pages 104–115, 2012.

2 Julius Richard Büchi and Lawrence H. Landweber. Solving sequential conditions by finite-state strategies. *Transactions of the American Mathematical Society*, 138:295–311, 1969.

3 Thomas Colcombet. Fonctions régulières de coût. Habilitation thesis, Université Paris Diderot—Paris 7, 2013.

4 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.

5 Thomas Colcombet and Christof Löding. The non-deterministic Mostowski hierarchy and distance-parity automata. In *ICALP (2)*, pages 398–409, 2008.

6 Jacques Duparc and Filip Murlak. On the topological complexity of weakly recognizable tree languages. *Fundamentals of computation theory*, 2007.

7 Alessandro Facchini and Henryk Michalewski. Deciding the Borel complexity of regular tree languages. In *CiE 2014*, pages 163–172, 2014.

8 Alessandro Facchini, Filip Murlak, and Michał Skrzypczak. Index problems for game automata. *ACM Trans. Comput. Log.*, 17(4):24:1–24:38, 2016.

9 Tomasz Gogacz, Henryk Michalewski, Matteo Mio, and Michał Skrzypczak. Measure properties of game tree languages. In *MFCS*, pages 303–314, 2014.

10 Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research*, volume 2500 of *Lecture Notes in Computer Science*. Springer, 2002.

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

12 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.

13 Orna Kupferman and Moshe Y. Vardi. The weakness of self-complementation. In *STACS*, pages 455–466, 1999.

14 Christof Löding. Logic and automata over infinite trees. Habilitation thesis, RWTH Aachen, Germany, 2009.

15 Satoru Miyano and Takeshi Hayashi. Alternating finite automata on omega-words. *Theor. Comput. Sci.*, 32:321–330, 1984.

16 Filip Murlak. The Wadge hierarchy of deterministic tree languages. *Logical Methods in Computer Science*, 4(4), 2008.

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

18 Damian Niwiński and Igor Walukiewicz. Deciding nondeterministic hierarchy of deterministic tree automata. *Electr. Notes Theor. Comput. Sci.*, 123:195–208, 2005.

19 Dominique Perrin and Jean-Éric Pin. *Infinite Words: Automata, Semigroups, Logic and Games*. Elsevier, 2004.

20 Michael Oser Rabin. Decidability of second-order theories and automata on infinite trees. *Trans. of the American Math. Soc.*, 141:1–35, 1969.

21 Michael Oser Rabin and Dana Scott. Finite automata and their decision problems. *IBM Journal of Research and Development*, 3(2):114–125, April 1959.

22 Michał Skrzypczak. *Descriptive Set Theoretic Methods in Automata Theory - Decidability and Topological Complexity*, volume 9802 of *Lecture Notes in Computer Science*. Springer, 2016.

23 Michał Skrzypczak and Igor Walukiewicz. Deciding the topological complexity of Büchi languages. In *ICALP (2)*, pages 99:1–99:13, 2016.

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

25 Wolfgang Thomas. Languages, automata, and logic. In *Handbook of Formal Languages*, pages 389–455. Springer, 1996.

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

27 Boris A. Trakhtenbrot. Finite automata and the monadic predicate calculus. *Siberian Mathematical Journal*, 3(1):103–131, 1962.