A Dichotomy Theorem for Ordinal Ranks in MSO

Damian Niwiński
Institute of Informatics, University of Warsaw, Poland

Paweł Parys
Institute of Informatics, University of Warsaw, Poland

Michał Skrzypczak
Institute of Informatics, University of Warsaw, Poland

Abstract

We focus on formulae $\exists X.\, \varphi(\vec{Y}, X)$ of monadic second-order logic over the full binary tree, such that the witness X is a well-founded set. The ordinal rank $\mathrm{rank}(X) < \omega_1$ of such a set X measures its depth and branching structure. We search for the least upper bound for these ranks, and discover the following dichotomy depending on the formula φ. Let η_φ be the minimal ordinal such that, whenever an instance $\vec{Y}$ satisfies the formula, there is a witness X with $\mathrm{rank}(X) \leq \eta_\varphi$. Then η_φ is either strictly smaller than ω^2 or it reaches the maximal possible value ω_1. Moreover, it is decidable which of the cases holds. The result has potential for applications in a variety of ordinal-related problems, in particular it entails a result about the closure ordinal of a fixed-point formula.

2012 ACM Subject Classification Theory of computation → Automata over infinite objects; Theory of computation → Tree languages

Keywords and phrases dichotomy result, limit ordinal, countable ordinals, nondeterministic tree automata

Digital Object Identifier 10.4230/LIPIcs.STACS.2025.69

Related Version *Full Version*: `https://arxiv.org/abs/2501.05385`

Funding All authors supported by the National Science Centre, Poland (grant no. 2021/41/B/ST6/03914).

Acknowledgements The authors would like to thank Marek Czarnecki for preliminary discussions on related subjects.

1 Introduction

The concept of well-founded relation plays a central role in foundations of mathematics. It gives rise to ordinal numbers, which underlie the basic results in set theory, for example that any two sets can be compared in cardinality. Well-foundedness is no less important in the realm of computer science, where it often underlies the proofs of *termination* of nondeterministic processes, especially when no efficient bound on the length of a computation is known. In such cases, the complexity of possible executions is usually measured using an ordinal called *rank*. Such a rank can be seen as a measure of the *depth* of the considered partial order, taking into account suprema of lengths of possible descending chains. Estimates on a rank can provide upper-bounds on the computational complexity of the considered problem [26].

In this work, we adopt the perspective of mathematical foundations of program verification and model-checking. We focus on the monadic second-order logic (MSO) interpreted in the infinite binary tree (with the left and right successors as the only non-logical predicates), which is one of the reference formalisms in the area [29]. The famous Rabin Tree Theorem [25] established its decidability, but – half a century after its introduction – the theory is still an object of study. On one hand, it has led to numerous extensions, often shifting the decidability

42nd International Symposium on Theoretical Aspects of Computer Science (STACS 2025).
Editors: Olaf Beyersdorff, Michał Pilipczuk, Elaine Pimentel, and Nguyễn Kim Thắng;
Article No. 69; pp. 69:1–69:18
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

result far beyond the original theory (see e.g. [4, 24]). On the other hand, a number of natural questions regarding Rabin's theory remain still open, including a large spectrum of *simplification* problems. For example, we still do not know whether we can decide if a given formula admits an equivalent form with all quantifiers restricted to finite sets. Similar questions have been studied in related formalisms like μ-calculus or automata; for example if we can effectively minimise the Mostowski index of a parity tree automaton [11, 15], or the $\mu\nu$-alternation depth of a μ-calculus formula [5].

On positive side, some decidability questions have been solved by reduction to the original theory. For example, it has been observed [22] that for a given formula $\varphi(\vec{X})$, the *cardinality* of the family of tuples of sets $\vec{X}$ satisfying $\varphi(\vec{X})$ can be computed; this cardinality can be either finite, $\aleph_0$, or $\mathfrak{c}$. Later on, Bárány, Kaiser, and Rabinovich [3] proved a more general result: they studied *cardinality quantifiers* $\exists^{\geq\kappa} X.\, \varphi(\vec{Y}, X)$, stating that there are at least κ distinct sets X satisfying $\varphi(\vec{Y}, X)$, and showed that these quantifiers can be expressed in the standard syntax of MSO; thus the extended theory remains decidable.

In the present work, instead of asking *how many* sets X witness to the formula $\exists X.\, \varphi(\vec{Y}, X)$, we ask how *complex* these witnesses must be in terms of their depth-and-branching structure. A set X of nodes of a tree is *well-founded* if it contains no infinite chain with respect to the descendant order. In this case, a countable ordinal number $\mathrm{rank}(X)$ is well defined (see Section 2 below); intuitively, the smaller $\mathrm{rank}(X)$, the simpler the set X is, in terms of its *branching structure*.

We consider formulae of the form $\exists X.\, \varphi(\vec{Y}, X)$, where φ is an arbitrary formula of MSO (it may contain quantifiers). We assume that, whenever the formula is satisfied for some valuation of variables $\vec{Y}$, the value of X witnessing the formula is a well-founded set. Note that well-foundedness of a set is expressible in MSO (it suffices to say that each branch contains only finitely many nodes in X), hence the requirement can be expressed within φ. For a fixed formula as above, we ask what is the minimal ordinal η_φ, such that the rank of a witness can be bounded by η_φ,

$$\eta_\varphi \stackrel{\text{def}}{=} \sup_{\vec{Y}} \min_{X} \ \mathrm{rank}(X), \tag{1.1}$$

where $\vec{Y}$ and X range, as expected, over the values satisfying $\varphi(\vec{Y}, X)$.

Since η_φ is a supremum of countable ordinals, its value is at most ω_1 (the least uncountable ordinal). It can be achieved, for example, by the formula "$X = Y$ *and* Y *is a well-founded set*", as there are well-founded sets of arbitrarily large countable ranks. On the other hand, for each pair of natural numbers (k, l), one can construct a formula φ with rank $\eta_\varphi = \omega \cdot k + l$ in an analogous way to Czarnecki [12], see Lemma 9.4 below. The main result of this work shows that no other ordinals can be obtained:

▶ **Theorem 1.1.** *For any formula $\exists X.\, \varphi(\vec{Y}, X)$ as above, the ordinal η_φ is either strictly smaller than ω^2 or equal to ω_1. Moreover, it can be effectively decided which of the cases holds. In the former case, it is possible to compute a number $N \in \mathbb{N}$ such that $\eta_\varphi < \omega \cdot N$.*

We also show that, in contrast to the aforementioned cardinality quantifiers, the property that $\mathrm{rank}(X)$ is smaller than ω^2 cannot be expressed directly in MSO (see Corollary 9.3).

The proof of Theorem 1.1 develops the game-based technique used previously to characterise certain properties of MSO-definable tree languages (see e.g. [9, 28]). Each application of this technique requires a specific game, designed in a way which reflects the studied property. The game should have a finite arena and be played between two perfectly-informed players $\exists$ and $\forall$. The winning condition of the game is given by a certain ω-regular set.

Then, the seminal result of Büchi and Landweber [8] yields that the game is determined and the winner of this game can be effectively decided. The construction of the game is such that a winning strategy of each of the players provides a witness of either of the considered possibilities: if $\exists$ wins then $\eta_\varphi = \omega_1$ and if $\forall$ wins then $\eta_\varphi < \omega \cdot N$ for some computable N. The crucial difficulty of this approach lies in a proper definition of the game, so that both these implications actually hold.

Related work

In the context of μ-calculus, one asks how many iterations are needed to reach a fixed point; this aspect concerns complexity of model checking (cf. [7, 14]), as well as expressive power of the logic (cf. [7, 6]). Recall that, in an infinite structure, a least fixed point $\mu X.\, F(X)$ is in general reached in a transfinite number of iterations: $\emptyset, F(\emptyset), \ldots, F^\alpha(\emptyset), \ldots$, where, for a limit ordinal α, $F^\alpha(\emptyset) = \bigcup_{\xi<\alpha} F^\xi(\emptyset)$. It is therefore natural to ask if, for a given formula, one can effectively find a *closure ordinal* η_F, such that, in any model, the fixed point can be reached in η_F, but in general not less, iterations. Fontaine [18] effectively characterised the formulae such that in each model the fixed point is reached in a finite number of steps. Czarnecki [12] observed that some formulae have no closure ordinals but, for each ordinal $\eta < \omega^2$, there is a formula whose closure ordinal is η. He also raised the following question.

▶ **Question 1.2** (Czarnecki [12]). *Is there a μ-calculus formula of the form $\mu X.\, F(X)$ that has a closure ordinal $\eta_F \geq \omega^2$?*

Gouveia and Santocanale [19] exhibited an example of a formula with an essential alternation of the least and greatest fixed-point operators whose closure ordinal is ω_1; clearly this limit can be achieved only in uncountable models. In general, it remains open whether a formula $\mu X.\, F(X)$ of the μ-calculus may have a *countable* closure ordinal $\eta_F \geq \omega^2$. Afshari and Leigh [2] claimed a negative answer for formulas of the alternation-free fragment of μ-calculus; however, as the authors have later admitted [1], the proof contained some gaps. In a recent paper [1], Afshari, Barlucchi, and Leigh update the proof and extend the result to formulae of the so-called Σ-*fragment* of the μ-calculus. More specifically, the authors consider systems of equations $X_1 = F_1(X_1, \ldots, X_n), \ldots, X_n = F_n(X_1, \ldots, X_n)$, where the formulae F_i may contain closed sub-formulae of the full μ-calculus, but the variables X_i do not fall in the scope of any fixed-point operators. The authors show that if a countable number of iterations η suffice to reach the least fixed point of such a system in any Kripke frame then $\eta < \omega^2$. (Note that, by Bekič principle, each component of such a solution can be expressed by a single formula with the nested μ-operators.)

As a direct consequence of our result, we obtain an alternative proof for formulae of the form $\mu X.\, F(X)$, where X does not fall in the scope of any fixed-point operator, but arbitrary closed subformulae may appear in F. (This corresponds to the case of $n = 1$ in the equation system of [1].) We show that no such formula has a closure ordinal η_F satisfying $\omega^2 \leq \eta_F < \omega_1$, and moreover, it can be decided whether $\eta_F < \omega^2$; see Section 10. This is achieved via a well-known reduction of the μ-calculus to the MSO theory of the binary tree, based on the tree-model property [7], and a natural encoding of a tree model.

In the studies of topological complexity of MSO-definable tree languages, it has been also observed that dichotomies may help to solve decision problems. An open problem related to the aforementioned question of definability with finite-set quantifiers, is whether we can decide if a tree language belongs to the Borel hierarchy (in general, it need not). A positive answer is known if a tree language is given by a deterministic parity automaton [23], based on the following dichotomy: such a language is either Π^1_1-complete (very hard), or on the level Π^0_3 (relatively low) of Borel hierarchy. Skrzypczak and Walukiewicz [28] gave a proof in the case when a tree language is given by a non-deterministic Büchi tree automaton, inspired by a rank-related dichotomy conjectured in the previous work of the first author [27, Conjecture 4.4]. There, an ordinal has been associated with each Büchi tree automaton, and it turns out (in view of [28]) that this ordinal either equals ω_1 or is smaller than ω^2, in which case the tree language is Borel. It should be mentioned that a procedure to decide if a Büchi definable tree language is weakly definable was given earlier by Colcombet et al. [10].

Relation between ordinals and automata has been also considered in the studies of automatic structures. In particular, Delhommé [13] showed that automatic ordinals are smaller than ω^ω while tree-automatic ordinals (defined in terms of automata on finite trees) can go higher, but not above ω^{ω^ω}. Later, Finkel and Todorcevic [17] showed that ω-tree automatic ordinals are smaller than ω^{ω^ω} as well. These results may appear in contrast with our more restrictive bound of ω^2; however, representation by automatic structures is in general more powerful than expressibility in MSO, so the two approaches are not directly related.

2 Basic notions

$\mathbb{N} = \{0, 1, 2, \ldots\}$ denotes the set of natural numbers. We use standard notation for ordinal numbers, with 0 being the least ordinal number, ω being the least infinite ordinal, and ω_1 the least uncountable ordinal. Although ω and $\mathbb{N}$ coincide as sets, we distinguish the two notations to emphasise the perspective.

Words. An alphabet A is any finite non-empty set, whose elements are called *letters*. A *word* over A is a finite sequence of letters $w = w_0 \cdots w_{n-1}$, with $w_i \in A$ for $i < n$, and n being the *length* of w. The empty word, denoted ε, is the unique word of length 0. By uv we denote the *concatenation* of words $u = u_0 \cdots u_{n-1}$ and $v = v_0 \cdots v_{m-1}$, that is, $uv = u_0 \cdots u_{n-1} v_0 \cdots v_{m-1}$.

A^* denotes the set of all finite words over the alphabet A, while A^ω denotes the set of all ω-words over A, that is, functions $\alpha \colon \mathbb{N} \to A$. If $\alpha \in A^\omega$ is an infinite word and $n \in \mathbb{N}$ then by $\alpha{\restriction}_n$ we denote the finite word consisting of the first n letters of α, that is, $\alpha_0 \cdots \alpha_{n-1} \in A^*$.

Trees. Let $\mathtt{L}$ and $\mathtt{R}$ denote two distinct letters called *directions*. For a direction $d \in \{\mathtt{L}, \mathtt{R}\}$, the *opposite direction* is denoted $\bar{d} \neq d$. Words over the alphabet $\{\mathtt{L}, \mathtt{R}\}$ are called *nodes*, with the empty word ε often called the *root*. An ω-word $\alpha \in \{\mathtt{L}, \mathtt{R}\}^\omega$ is called an *infinite branch*. A node u is called the *parent* of its *children* $u\mathtt{L}$ and $u\mathtt{R}$.

A (full infinite binary) *tree* over an alphabet A is a function $t \colon \{\mathtt{L}, \mathtt{R}\}^* \to A$, assigning letters $t(u) \in A$ to nodes $u \in \{\mathtt{L}, \mathtt{R}\}^*$. The set of all trees over A is denoted Tr_A. A *subtree* of a tree $t \in \mathrm{Tr}_A$ in a node $u \in \{\mathtt{L}, \mathtt{R}\}^*$ is the tree $t{\restriction}_u$ over A defined by taking $t{\restriction}_u(v) = t(uv)$.

Automata. Instead of working with formulae of MSO, we use another equivalent formalism, namely non-deterministic parity tree automata [29]. An *automaton* is a tuple $\mathcal{A} = (A, Q, q_{\mathrm{I}}, \Delta, \Omega)$, where A is an alphabet, Q is a finite set of *states*, $q_{\mathrm{I}} \in Q$ is an *initial state*, $\Delta \subseteq Q \times A \times Q \times Q$ is a *transition relation*, and $\Omega \colon Q \to \{i, \ldots, j\} \subseteq \mathbb{N}$ is a *priority mapping*. A *run* of an automaton $\mathcal{A}$ from a state $q \in Q$ over a tree $t \in \mathrm{Tr}_A$ is a tree $\rho \in \mathrm{Tr}_Q$ such that $\rho(\varepsilon) = q$ and for every node $u \in \{\mathtt{L}, \mathtt{R}\}^*$ the quadruple

$$\big(\rho(u), t(u), \rho(u\mathtt{L}), \rho(u\mathtt{R})\big),$$

is a transition of $\mathcal{A}$ (i.e., belongs to Δ).

A sequence of states $(q_0, q_1, \ldots) \in Q^\omega$ is *accepting* if $\limsup_{n\to\infty} \Omega(q_n)$ is an even natural number. A run ρ is *accepting* if for every infinite branch $\alpha \in \{\mathtt{L},\mathtt{R}\}^\omega$ the sequence of states $\big(\rho(\alpha{\restriction}_n)\big)_{n\in\mathbb{N}}$ that appear in ρ on α is accepting.

A tree $t \in \mathrm{Tr}_A$ is *accepted* by an automaton $\mathcal{A}$ from a state $q \in Q$ if there exists an accepting run ρ of $\mathcal{A}$ from the state q over t. The *language* of $\mathcal{A}$, denoted $\mathrm{L}(\mathcal{A}) \subseteq \mathrm{Tr}_A$, is the set of trees which are accepted by the automaton from the initial state q_I. A language $L \subseteq \mathrm{Tr}_A$ is *regular* if it is $\mathrm{L}(\mathcal{A})$ for some automaton $\mathcal{A}$.

We now recall the famous theorem of Rabin, which allows us to transform MSO formulae into equivalent tree automata.

▶ **Theorem 2.1** ([25], see also [29])**.** *For every MSO formula $\varphi(X_0, \ldots, X_{n-1})$ the set of valuations satisfying φ is a regular language over the alphabet $\{0,1\}^n$.*

We say that an automaton is *pruned* if every state $q \in Q$ appears in some accepting run, that is, there exists an accepting run ρ of $\mathcal{A}$ over a tree t such that $\rho(u) = q$ for some node $u \in \{\mathtt{L},\mathtt{R}\}^*$. Note that every automaton can effectively be pruned, without affecting its language, by detecting and removing states that do not appear in any accepting run.

Games. We use the standard framework of two-player games of infinite duration (see e.g., [8, 16]). The arena of such a game is given as a graph with both players having perfect information about the current position. The winning condition of the game is given by a language of infinite plays won by one of the players.

Ranks. Recall that an ordinal η can be seen as the linearly ordered set of all ordinals smaller than η. Given a set $X \subseteq \{\mathtt{L},\mathtt{R}\}^*$, a *counting function* is a function $\mathfrak{C}\colon X \to \eta$ such that $\mathfrak{C}(u) < \mathfrak{C}(v)$ whenever $u \in X$ is a proper descendant of $v \in X$ (such a function exists for some η whenever X is well-founded). The *rank* of a well-founded set X is the least ordinal η for which a counting function from X to η exists.

Taking the automata-based perspective, instead of working with monadic variables $\vec{Y}$ and X, we consider labellings of the tree by certain finite alphabets. In particular, a set of nodes $X \subseteq \{\mathtt{L},\mathtt{R}\}^*$ can be identified with its characteristic function, that is, a tree $x \in \mathrm{Tr}_{\{0,1\}}$ over the alphabet $\{0,1\}$. Such a tree is *well-founded* if no infinite branch contains infinitely many nodes labelled by 1. The set of well-founded trees is denoted $\mathrm{WF} \subseteq \mathrm{Tr}_{\{0,1\}}$. Likewise, we define the *rank* of $x \in \mathrm{WF}$, denoted $\mathrm{rank}(x)$, as the least η for which there is a counting function from the set of 1-labelled nodes of x to η. The considered rank is analogous with the standard rank of well-founded trees (e.g., [21, Section 2.E]).

▶ **Example 2.2.** The tree $x_0 \in \mathrm{Tr}_{\{0,1\}}$ with all nodes labelled by 0 has rank 0.

Consider a tree x_ω where a node v has label 1 if $v = \mathtt{R}^i\mathtt{L}^j$ with $1 \leq j \leq i$. It is a *comb*: the rightmost branch is labelled by zeros, and below its i-th node we have a tooth of i nodes labelled by ones, going left. The rank of x_ω is ω.

Such combs can be nested: suppose that $x_{\omega\cdot 2}$ starts analogously to x_ω, but below every tooth we again insert x_ω (i.e., $x_{\omega\cdot 2}{\restriction}_{\mathtt{R}^i\mathtt{L}^{i+1}} = x_\omega$ for every i); then $\mathrm{rank}(x_{\omega\cdot 2}) = \omega\cdot 2$. Repeating this, we can insert $x_{\omega\cdot n}$ below every tooth of x_ω, and obtain $x_{\omega\cdot(n+1)}$ of rank $\omega\cdot(n{+}1)$, for every $n \in \mathbb{N}$.

Then, we can place every $x_{\omega\cdot n}$ at node $\mathtt{R}^n\mathtt{L}$, below a 0-labelled rightmost branch; the resulting tree has rank ω^2. In a similar manner we can create a tree having rank equal to any countable ordinal η.

3 Problem formulation

We begin by formulating the problem under consideration. Instead of working with a formula $\varphi(\vec{Y}, X)$, we assume that A is some alphabet and Γ is a regular language over the alphabet $A \times \{0,1\}$. We identify a tree τ over $A \times \{0,1\}$ with a pair (t,x), where $t \in \mathrm{Tr}_A$ and $x \in \mathrm{Tr}_{\{0,1\}}$ such that $\tau(u) = \big(t(u), x(u)\big)$ for every node $u \in \{\mathtt{L},\mathtt{R}\}^*$. Thus, $\Gamma \subseteq \mathrm{Tr}_{A\times\{0,1\}}$ can be seen as a relation whose elements are pairs (t,x). We additionally require that whenever $(t,x) \in \Gamma$ then x is well-founded, which means that Γ (treated as a relation) is contained in $\mathrm{Tr}_A \times \mathrm{WF}$. We say that such a relation is *regular* if it is regular as a language over $A \times \{0,1\}$.

Let $\pi_A(\Gamma)$ be the projection of Γ onto the A coordinate, that is, the set of those trees $t \in \mathrm{Tr}_A$ for which there exists a (necessarily well-founded) tree $x \in \mathrm{Tr}_{\{0,1\}}$ such that $(t,x) \in \Gamma$. Similarly, for a tree $t \in \mathrm{Tr}_A$ by Γ_t we denote the *section* of Γ over the tree t, that is, the set $\{x \in \mathrm{Tr}_{\{0,1\}} \mid (t,x) \in \Gamma\}$.

The following definition is just a reformulation of Formula (1.1) in terms of a relation Γ.

▶ **Definition 3.1.** *The* closure ordinal *of a relation* $\Gamma \subseteq \mathrm{Tr}_A{\times}\mathrm{WF} \subseteq \mathrm{Tr}_{A\times\{0,1\}}$ *(or of an automaton recognising it) is defined as*

$$\eta_\Gamma \stackrel{\text{def}}{=} \sup_{t\in\pi_A(\Gamma)} \ \min_{x\in\Gamma_t} \ \mathrm{rank}(x).$$

▶ **Example 3.2.** Consider the following automaton $\mathcal{A}$ over the alphabet $\{\mathsf{b},\mathsf{c}\} \times \{0,1\}$. Its states are $p_0, q_1, q_2, q_3, r_0, r_1$ with q_1 being initial, where the subscript provides the priority of a state (i.e., $\Omega(p_i) = \Omega(q_i) = \Omega(r_i) = i$). The transitions are, for all $i \in \{0,1\}$, $j \in \{1,2,3\}$, and $a \in \{\mathsf{b},\mathsf{c}\}$,

$$\begin{array}{lll}
(p_0,(a,0),p_0,p_0), & (q_j,(\mathsf{c},0),q_3,p_0), & (q_j,(\mathsf{c},0),p_0,r_0),\\
(q_j,(\mathsf{b},0),q_2,p_0), & (q_j,(\mathsf{c},0),p_0,q_3), & (r_i,(\mathsf{b},i),r_1,r_0),\\
(q_j,(\mathsf{b},0),p_0,q_1), & (q_j,(\mathsf{c},0),r_0,p_0), & (r_i,(\mathsf{c},0),p_0,p_0).
\end{array}$$

In this example, we should see c-labelled nodes as separators, splitting the whole tree into b-labelled fragments. Let us see when a pair $(t,x) \in \mathrm{Tr}_{\{\mathsf{b},\mathsf{c}\}\times\{0,1\}}$ can be accepted. Note first that $\mathcal{A}$ accepts (t,x_0) from p_0 for every tree t, and for x_0 having all nodes labelled by 0. Next, observe that states q_j become aligned along a single branch, either finite or infinite. If in t there is a branch that infinitely often goes left, but visits only finitely many c-labelled nodes, then we can align the q_j states along this branch, and (t,x_0) will be accepted. Indeed, just below every c we have q_3 (state of large odd priority; has to occur finitely often); below every b, if we go left we have q_2, and if we go right we have q_1 (so the parity condition requires infinitely many left-turns). Another possibility is that the branch with states q_j is finite, and below some c-labelled node the state changes to r_0. Now the run sends r_1 to every left child, and r_0 to every right child; hence every left child in x should have label 1, and every right child – label 0. We can continue the zone of states r_i until reaching a node labelled by c; such a node allows us to change the state into p_0 and accept anything below. The acceptance condition requires that there are only finitely many states r_1 (hence also nodes with label 1 in x) on every branch; the tree x is necessarily well-founded.

This determines the optimal rank of a witness x for a tree t. Namely, if in t there is a branch that infinitely often goes left, but visits only finitely many c-labelled nodes, we have a witness of rank 0. Otherwise, we should consider every zone of b-labelled nodes in t, surrounded by c-labelled nodes; consider x having 1 in every left child in that zone; and take the minimum of ranks of such trees x, over all choices of zones (not including the topmost zone, above the first c on a branch). Thus, every witness of a tree t has rank at least η if and only if every such zone results in rank at least η, and the former case does not hold. Such a tree exists for every $\eta < \omega_1$, so the closure ordinal of $\mathcal{A}$ is ω_1.

4 The dichotomy game

We now move to the definition of the game $\mathcal{G}_{\mathcal{A}}$ designed to decide the dichotomy from Theorem 1.1.

Note that for every regular relation $\Gamma \subseteq \mathrm{Tr}_{A\times\{0,1\}}$ there exists another regular relation $\Gamma' \subseteq \mathrm{Tr}_{A\times\{0,1\}}$ with the same closure ordinal, but such that $\pi_A(\Gamma')$ is the set of all trees over A. To see this, it is enough to take $\Gamma' = \Gamma \cup \{(t,x') \mid x' \in \mathrm{WF} \wedge \neg\exists x.(t,x) \in \Gamma\}$. Then $\min_{x\in\Gamma_t} \mathrm{rank}(x) = \min_{x\in\Gamma'_t} \mathrm{rank}(x)$ for $t \in \pi_A(\Gamma)$ and $\min_{x\in\Gamma'_t} \mathrm{rank}(x) = 0$ for $t \notin \pi_A(\Gamma)$.

Towards the proof of Theorem 1.1, we consider some relation $\Gamma \subseteq \mathrm{Tr}_{A\times\{0,1\}}$ such that $\Gamma \subseteq \mathrm{Tr}_A \times \mathrm{WF}$ and $\pi_A(\Gamma) = \mathrm{Tr}_A$. Reformulating Theorem 1.1, we need to show that either $\eta_\Gamma < \omega \cdot N$ for some $N \in \mathbb{N}$, or $\eta_\Gamma = \omega_1$, and we can effectively decide which case holds. We assume that Γ is given by a pruned (i.e., all states appear in some accepting run) automaton $\mathcal{A} = (A, Q, q_\mathrm{I}, \Delta, \Omega)$.

First, a *side* is a symbol $s \in \{\mathsf{T}, \mathsf{R}\}$ (which stands for “trunk” and “reach”), and a *mode* is a symbol $m \in \{①, ②\}$ (which stands for non-branching and binary-branching).

A quadruple $(\delta, s, m, d) \in \Delta \times \{\mathsf{T}, \mathsf{R}\} \times \{①, ②\} \times \{\mathtt{L}, \mathtt{R}\}$ such that $s = \mathsf{R}$ implies $m = ①$ (i.e., $m = ②$ is allowed only for $s = \mathsf{T}$) is called a *selector for* (δ, s), where δ is of the form $\big(q, (a, i), q_\mathtt{L}, q_\mathtt{R}\big)$ with $a \in A$, $i \in \{0, 1\}$. Such a selector *agrees with* a direction $d' \in \{\mathtt{L}, \mathtt{R}\}$ if either $m = ②$ or $d' = d$ (i.e., both directions d' are fine if $m = ②$, but if $m = ①$ then we require $d' = d$). The *output side* of a selector (δ, s, m, d) in direction d', where $\delta = (q, (a, i), q_\mathtt{L}, q_\mathtt{R})$, is

- T if $s = \mathsf{T}$ and $d' = d$, or $s = \mathsf{R}$ and $i = 1$, and
- R otherwise: if $s = \mathsf{T}$ and $d' \neq d$, or $s = \mathsf{R}$ and $i = 0$.

A *state-flow* is a triple $\big((q, s), m, (q', s')\big)$, where $q, q' \in Q$, $s, s' \in \{\mathsf{T}, \mathsf{R}\}$, and $m \in \{①, ②\}$. A *flow* μ is a set of state-flows. The set $\big\{(q', s') \mid \big((q, s), m, (q', s')\big) \in \mu\big\}$ is called the *image* of μ. Note that the number of all possible state-flows, hence also of all possible flows, is finite.

Given a flow μ, we say that a flow $\bar{\mu} \subseteq \mu$ is a *back-marking* of μ if for every pair (q', s') in the image of μ, precisely one state-flow leading to (q', s') belongs to $\bar{\mu}$. This state flow is called *back-marked* for (q', s').

Given a sequence of flows, $\mu_1, \mu_2, \ldots$ we can define their *composition* as the graph with vertices $(q, s, n) \in Q \times \{\mathsf{T}, \mathsf{R}\} \times \mathbb{N}$ and with a directed edge from (q, s, n) to $(q', s', n{+}1)$ labelled by m for every state-flow $\big((q, s), m, (q', s')\big) \in \mu_{n+1}$, $n \in \mathbb{N}$. Note that in a flow there may be two state-flows with the same pairs (q, s) and (q', s'), with modes $m = ①$ and $m = ②$, leading to two parallel edges in this graph.

Assume that we are given a set $Q' \subseteq Q$ and a letter $a \in A$. By $\Delta_a(Q')$ we denote the set of transitions of the form $\big(q, (a, i), q_\mathtt{L}, q_\mathtt{R}\big)$ with $q \in Q'$.

Finally, for two sets of states $T, R \subseteq Q$ we denote $(\!| T, R |\!) \stackrel{\text{def}}{=} (T{\times}\{\mathsf{T}\}) \cup (R{\times}\{\mathsf{R}\})$. Note that every subset of $Q \times \{\mathsf{T}, \mathsf{R}\}$ can be uniquely represented as $(\!| T, R |\!)$ for some $T, R \subseteq Q$.

We can now move to the definition of the crucial game $\mathcal{G}_{\mathcal{A}}$, used to prove the desired dichotomy. The positions of $\mathcal{G}_{\mathcal{A}}$ are of the form $(\!| T, R |\!) \subseteq Q \times \{\mathsf{T}, \mathsf{R}\}$ (formally, one also needs additional auxiliary positions to represent the situation between particular steps of a round; we do not refer to these positions explicitly). The initial position is $(\!| \{q_\mathrm{I}\}, \emptyset |\!)$, where q_I is the initial state of the automaton $\mathcal{A}$. The consecutive steps in a round $n \in \mathbb{N}$ from a position $(\!| T_n, R_n |\!)$ are as follows:

1. $\forall$ declares a subset $T'_n \subseteq T_n$.
2. $\exists$ declares a letter $a_n \in A$.
3. $\exists$ declares a set F_n of selectors, containing one selector for each $(\delta, s) \in (\Delta_{a_n}(T'_n) \times \{\mathsf{T}\}) \cup (\Delta_{a_n}(R_n) \times \{\mathsf{R}\})$.
4. $\forall$ declares a direction $d_{n+1} \in \{\mathtt{L}, \mathtt{R}\}$.
5. We define a flow μ_{n+1} as the set containing the state-flows $\big((q, s), m, (q_{d_{n+1}}, s')\big)$ for each selector $(\delta, s, m, d) \in F_n$ that agrees with direction d_{n+1}, where s' is the output side of the selector in the direction d_{n+1}, and $\delta = (q, (a_n, i), q_{\mathtt{L}}, q_{\mathtt{R}})$ (so q is the source state of δ and $q_{d_{n+1}}$ is the state sent by δ in direction d_{n+1}).
6. $\forall$ declares some back-marking $\bar{\mu}_{n+1}$ of the flow μ_{n+1}.

The new position of the game, $(\!|T_{n+1}, R_{n+1}|\!)$, is the image of the flow μ_{n+1}.

Given a play Π of the game, the winning condition for $\exists$ is the disjunction **A)** $\vee$ **B)** of the following parts.

A) In the graph obtained as a composition of the back-markings $\bar{\mu}_1, \bar{\mu}_2, \ldots$ there exists a path which infinitely many times changes sides between T and R.

B) In the graph obtained as a composition of the flows $\mu_1, \mu_2, \ldots$ every infinite path either is rejecting or contains infinitely many state-flows of mode ②.

Above, while saying that a path going through $(q_0, s_0, 0), (q_1, s_1, 1), \ldots$ is rejecting, we mean that the sequence of states $(q_n)_{n\in\mathbb{N}}$ is rejecting, that is, $\limsup_{n\to\infty} \Omega(q_n)$ is odd.

▶ Remark 4.1. The arena of the game $\mathcal{G}_{\mathcal{A}}$ is finite and the winning condition defined above is ω-regular. Thus, the theorem of Büchi and Landweber [8] applies: one can effectively decide the winner of $\mathcal{G}_{\mathcal{A}}$. Moreover, there exists a computable bound M such that whoever wins $\mathcal{G}_{\mathcal{A}}$, can win using a finite-memory strategy that uses at most M memory states.

The following proposition formalises the relation between $\mathcal{G}_{\mathcal{A}}$ and Theorem 1.1.

▶ **Proposition 4.2.** *Assume that $\mathcal{A}$ is a pruned automaton which recognises a relation $\Gamma \subseteq \mathrm{Tr}_A \times \mathrm{WF}$ such that $\pi_A(\Gamma) = \mathrm{Tr}_A$ (i.e., the projection of Γ is full). Then, we have the following two implications:*

1. *if player $\exists$ wins $\mathcal{G}_{\mathcal{A}}$ then $\eta_\Gamma = \omega_1$;*
2. *if player $\forall$ wins $\mathcal{G}_{\mathcal{A}}$ then $\eta_\Gamma < \omega \cdot N$, for a number $N \in \mathbb{N}$ computable based on the automaton $\mathcal{A}$.*

Depiction of a round of $\mathcal{G}_{\mathcal{A}}$. Figure 4.1 depicts an example of a round of the game $\mathcal{G}_{\mathcal{A}}$. First $\forall$ declares a subset $T'_n = \{q_0, q_1\}$ effectively removing the state q_2 from the T side. Note that some states might repeat on both sides. Once the letter $a_n = a$ is chosen by $\exists$, there are six possible transitions from the considered states: one from each of them, except the state q_1 which has two possible transitions, δ_1 over $(a, 0)$ and δ_2 over $(a, 1)$. $\exists$ declares a set of selectors

$$F_n = \Big\{(\delta_0, \mathsf{T}, ②, \mathtt{L}), (\delta_1, \mathsf{T}, ①, \mathtt{L}), (\delta_2, \mathsf{T}, ②, \mathtt{R}), (\delta_3, \mathsf{R}, ①, \mathtt{L}), (\delta_4, \mathsf{R}, ①, \mathtt{L}), (\delta_5, \mathsf{R}, ①, \mathtt{R})\Big\}.$$

Thus, the selectors for the transitions δ_0 and δ_2 are in the mode ②, while the remaining selectors are in the mode ①. Once $\forall$ chooses the direction $d_{n+1} = \mathtt{L}$, we gather into new sets T_{n+1} and R_{n+1} the states sent by the transitions in the direction $\mathtt{L}$. The transitions δ_0 and δ_1 provide their left states into T_{n+1}. However, in the case of transition δ_2 the direction of the selector is $\mathtt{R}$ while the direction chosen by $\forall$ is $\mathtt{L}$ and therefore the left state of this transition goes to R_{n+1}. Regarding the transitions on the R side, δ_3 provides its state to T_{n+1} because this transition is over $(a, 1)$. The transition δ_4 provides its state to R_{n+1} because

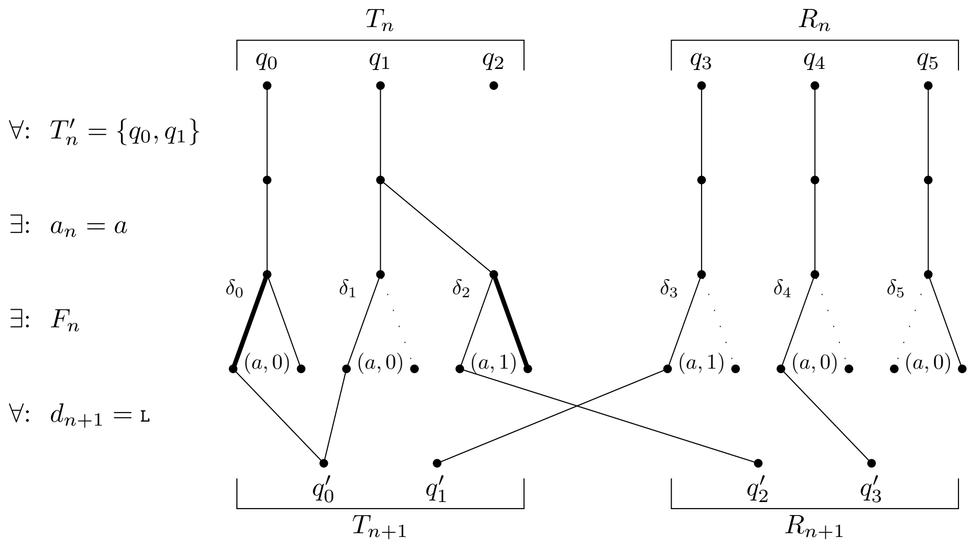

Figure 4.1 A depiction of a round of the game $\mathcal{G}_{\mathcal{A}}$.

this transition is over $(a, 0)$. Finally, the transition δ_5 does not provide its state anywhere, because its mode is ① and its direction is R, different than the direction L chosen by ∀. The states provided by the transitions δ_0 and δ_1 are both q'_0 and are therefore merged. This means that the final flow μ_{n+1} is

$$\Big\{\big((q_0,\mathsf{T}),②,(q'_0,\mathsf{T})\big),\big((q_0,\mathsf{T}),①,(q'_0,\mathsf{T})\big),\big((q_1,\mathsf{T}),②,(q'_2,\mathsf{R})\big),$$
$$\big((q_3,\mathsf{R}),①,(q'_1,\mathsf{T})\big),\big((q_4,\mathsf{R}),①,(q'_3,\mathsf{R})\big)\Big\}.$$

When ∀ chooses a back-marking $\bar{\mu}_{n+1}$ of the flow μ_{n+1}, he needs to make a choice which state-flow to select for the pair (q'_0, T) as there are two possible choices.

5 Intuitions

Let us explain the intuitions behind the game. First very generally: the game is designed so that ∃ wins when there are trees t requiring witnesses x of arbitrarily large ranks. Conversely, ∀ wins if there is a bound η such that every tree t has a witness x of rank at most η. But because this is a finite game with an ω-regular winning condition, if ∀ wins then he wins with a finite-memory winning strategy; from such a strategy we can deduce that the bound η is actually of the form $\omega \cdot N$ for some natural number N depending on the size of the memory of ∀.

Having the above in mind, let us discuss details of the game. The role of ∃ should be to show a tree t (whose all witnesses x have large rank), so we allow her to propose a label of a node in step 2. However, as usually in games, we do not continue in both children of the current node, but we rather ask ∀ to choose one direction (d_{n+1} in step 4), where ∃ has to continue creating the tree, and where ∀ thinks that it is impossible to continue.

Recall now that arbitrarily large countable ordinals can be obtained by alternately applying two operations in a well-founded way: add one, and take the supremum of infinitely many ordinals. Similarly, in a tree x of a large rank, we can always find a *comb*: an infinite branch – a trunk – such that from infinitely many nodes on the side of this trunk we can reach a node labelled by 1, below which the rank is again large (but slightly smaller). In these places we can repeat this process, that is, again find an analogous comb, and so on, obtaining a tree of nested combs that is well-formed but itself has a large rank. Obviously the converse holds as well: such a tree of nested combs having large rank can exist inside x only if x has a large rank. Let us also remark that in the case of a finite-memory strategy of $\forall$ we obtain that such combs in x can be nested at most $N \in \mathbb{N}$ times, which itself implies that the rank of x is at most $\omega \cdot N$.

Thus, in order to show that the constructed tree t allows only witnesses x of large rank, $\exists$ shows a nested comb structure. But this has to be done for every x such that $(t, x) \in \Gamma$, so, in a sense, for every run of $\mathcal{A}$ over (t, x) for some x. As usual, we cannot require from $\forall$ to choose a run on the fly, during the game; an interesting (even an accepting) run of $\mathcal{A}$ can only be fixed after the whole tree (i.e., the whole future of the play) is fixed. This means that during the game we have to trace all possible runs of $\mathcal{A}$. However, there are infinitely many runs, so we cannot do that. To deal with that, we keep track of all "interesting" states in the sets T_n, R_n, and we consider all possible transitions from them in step 3. This makes the situation of $\exists$ a bit worse: she has to make decisions based only on the current state, not knowing the past of the run (the same state may emerge after two different run prefixes). But it turns out that $\exists$ can handle that; in our proof this corresponds to positionality of strategies in an auxiliary game considered in Section 7.

Now, how exactly does $\exists$ show the nested comb structure? This is done via selectors proposed in the sets F_n. For states in T_n (i.e., on the "trunk" side) $\exists$ has to show in which direction the trunk continues. Moreover, $\exists$ has to show places where on side of the trunk we can reach label 1 followed by a nested comb; in these places $\exists$ plays mode ② ("branch"). If $\forall$ chooses a direction in which the trunk (as declared by $\exists$) continues, we trace the resulting state again on the T side. If he chooses the opposite direction, and the mode is ① (no branching here), we just stop tracing this run. But if $\forall$ chooses the non-trunk direction while the mode is ②, the resulting state ends up on the R ("reach") side. The role of $\exists$ is now to show a direction in which we can reach a node with label 1. If $\forall$ follows this direction, and the label of x is 0, we continue searching for label 1 on the R side. When label 1 is found, we put the resulting state on the T side; $\exists$ has to show a next comb. The **B)** part of the winning condition ensures for every accepting run that the trunk of each comb has infinitely many branching points (i.e., points with mode ②) and that on the R side we stay only for finitely many steps (as mode ② does not occur there). Note that by arbitrarily composing transitions we may obtain also rejecting runs; the **B)** condition does not require anything for them.

There is one more issue taken into account in the design of the game: $\exists$ should be obliged to produce the combs nested arbitrarily many times, but not infinitely many times. The number of *nestings* (i.e., of switches between sides T and R) is controlled by $\forall$. When he is satisfied with the nesting depth provided by $\exists$ for runs ending in some state, he can remove this step from the position in step 1, and let $\exists$ provide appropriate comb structures only from remaining states (we allow this removal only on the T side, but we could equally well allow it on the R side, or on both sides). The **A)** part of the winning condition obliges $\forall$ to indeed remove a state after seeing finitely many nestings. To see the usefulness of back-marking used to formulate this condition, consider a situation with two runs leading to some state: one with already many nestings of combs provided by $\exists$, and other where we are still on the trunk of the first comb. Because $\exists$ should provide many nested combs for all runs, in such a situation $\forall$ should still be able to analyse the latter run. To this end, he can select the latter run as the back-marked history of the considered state, and continue waiting for further nested combs, without worrying that he will lose by the **A)** part of the winning condition due to the former run.

▶ **Example 5.1.** Let us see how the game $\mathcal{G}_\mathcal{A}$ behaves for the automaton $\mathcal{A}$ from Example 3.2. Recall that the closure ordinal of $\mathcal{A}$ is ω_1, so $\exists$ should be able to win.

The strategy of $\exists$ from a position $(\!|T'_n, R_n|\!)$ is as follows. If $T'_n \cup R_n$ contains a state r_i, then $\exists$ plays letter b, otherwise letter c. Then $\exists$ proposes selectors: transitions δ originating from states q_j are handled by selectors (δ, s, m, d) with mode $m = ①$ and with direction d being such that δ sends state p_0 to the opposite direction $\bar{d}$ (i.e., $d = \mathtt{L}$ for $\delta = (q_j, (a, 0), q_2, p_0)$, etc.); transitions originating from r_i are handled by direction $d = \mathtt{R}$, and by mode $m = ②$ on the T side and mode $m = ①$ on the R side. As we argue below, p_0 never becomes an element of $T_n \cup R_n$, so transitions from p_0 need not to be handled.

The initial position is $(\!|\{q_1\}, \emptyset|\!)$. Here $\exists$ plays letter c and some selectors with mode ①, and $\forall$ chooses a direction. There are two selectors (δ, s, m, d) that agree with this direction, and they send there states q_3 and r_0. Thus the next position is $(\!|\{q_3, r_0\}, \emptyset|\!)$, reached by the flow $\{((q_1, \mathsf{T}), ①, (q_3, \mathsf{T})), ((q_1, \mathsf{T}), ①, (r_0, \mathsf{T}))\}$. This time $\exists$ plays letter b. If $\forall$ goes right, the new position is $(\!|\{q_1, r_0\}, \emptyset|\!)$, reached by the flow $\{((q_3, \mathsf{T}), ①, (q_1, \mathsf{T})), ((r_0, \mathsf{T}), ②, (r_0, \mathsf{T}))\}$; this position behaves identically to the previous one, as from q_1 and q_3 we have the same transitions. If $\forall$ goes left, the new position is $(\!|\{q_2\}, \{r_1\}|\!)$, reached by the flow $\{((q_3, \mathsf{T}), ①, (q_2, \mathsf{T})), ((r_0, \mathsf{T}), ②, (r_1, \mathsf{R}))\}$. From $(\!|\{q_2\}, \{r_1\}|\!)$ once again letter b is played. Note that the only transition from r_1 reading letter b on the first coordinate, reads 1 on the second coordinate. So, if $\forall$ goes right, the new position is back to $(\!|\{q_1, r_0\}, \emptyset|\!)$, reached by the flow $\{((q_2, \mathsf{T}), ①, (q_1, \mathsf{T})), ((r_1, \mathsf{R}), ①, (r_0, \mathsf{T}))\}$; if he goes left, he reaches $(\!|\{q_2\}, \emptyset|\!)$, which behaves like the initial position.

It is also possible that $\forall$ erases a state from the T side (i.e., plays T'_n being a proper subset of T_n). If state r_0 is erased, we end up in a position $(\!|\{q_j\}, \emptyset|\!)$, being like the initial position. We may also have positions $(\!|\{r_0\}, \emptyset|\!)$ and $(\!|\emptyset, \{r_1\}|\!)$, and flows $\{((r_0, \mathsf{T}), ②, (r_0, \mathsf{T}))\}$, $\{((r_0, \mathsf{T}), ②, (r_1, \mathsf{R}))\}$, $\{((r_1, \mathsf{R}), ①, (r_0, \mathsf{T}))\}$ between them. Finally, we may also reach $(\!|\emptyset, \emptyset|\!)$.

Note that in our example there is always only one state-flow leading to every pair (q, s); in consequence, every state-flow is back-marked.

Let us now check the winning condition. One possibility is that infinitely many letters c were played. In these moments no state r_i was present in the position, so the only infinite path in the composition of flows is the path going through appropriate q_j states. But after seeing every c this state was q_3, so the path is rejecting; part **B)** of the winning condition is satisfied (we may also have no infinite path, if the q_j state was removed by $\forall$, but then condition **B)** holds even more). The opposite case is that from some moment on, only letter b was played. We then also have an infinite path going, from some moment, through the r_i states (this path really exists: if $\forall$ removes the state r_0, then letter c is played). Note that whenever $\forall$ goes left, this path changes sides from T to R, and in the next round returns back to the T side. If this happens infinitely often, $\exists$ wins by condition **A)**. Otherwise, from some moment on $\forall$ constantly goes right. After that, the path going through the r_i states has all state-flows of mode ②, and the other path (if exists) remains in state q_1, so it is rejecting; condition **B)** is satisfied.

6 Soundness

We begin by proving Item 1 of Proposition 4.2: if $\exists$ wins $\mathcal{G}_\mathcal{A}$ then $\eta_\Gamma = \omega_1$. To this end, assume that $\exists$ wins the game and fix any winning strategy for her. For every countable ordinal $\eta > 0$, our goal is to construct a tree t such that $\mathrm{rank}(x) \geq \eta$ whenever $(t, x) \in \Gamma$. We do this by inductively unravelling the fixed strategy of $\exists$. During this process, we keep track of

- the current node $v \in \{\mathtt{L}, \mathtt{R}\}^*$,
- the current position $(\!|T_v, R_v|\!)$ of the game,
- a mapping κ_v, which assigns some ordinals $\leq \eta$ to all elements of the set $(\!|T_v, R_v|\!)$.

Initially $v = \varepsilon$, the position $(\!|T_\varepsilon, R_\varepsilon|\!)$ is the initial position of the game (i.e., $(\!|\{q_\mathrm{I}\}, \emptyset|\!)$), and $\kappa_\varepsilon = \{(q_\mathrm{I}, \mathsf{T}) \mapsto \eta\}$. Then, in every node v, the letter declared by $\exists$ provides a label of this node in t, and while moving to a child vd of v we trace a play in which $\forall$ declares the respective direction d. What remains is to declare the remaining choices of $\forall$ and say how the mapping κ_v is updated. The role of the ordinal $\kappa_v(q, s)$ is to provide a lower bound for ranks of witnesses x such that $(t{\restriction}_v, x)$ can be accepted by $\mathcal{A}$ from the state q. While going down the tree, this ordinal decreases whenever we change the side from T to R; when it becomes 0, $\forall$ removes the respective state from the set T_v in his move. The back-markings are declared by $\forall$ in a way that maximises the ordinals $\kappa_v(q, s)$ of the respective state-flows.

It remains to show that if $(t, x) \in \Gamma$ then $\mathrm{rank}(x) \geq \eta$. This is achieved by considering any accepting run ρ of $\mathcal{A}$ over (t, x). Since all possible transitions of $\mathcal{A}$ are taken into account in each round of $\mathcal{G}_\mathcal{A}$, one can trace the run ρ in the simulated plays of $\mathcal{G}_\mathcal{A}$. The policy of updating the mapping κ_v inductively assures that for every node v and currently traced side s_v we have $(\rho(v), s_v) \in (\!|T_v, R_v|\!)$ and $\mathrm{rank}(x{\restriction}_v) \geq \kappa_v(\rho(v), s_v)$. Moreover, this policy guarantees that condition **A)** is never satisfied in these plays. Thus, condition **B)** must hold, inductively guaranteeing the inequality on $\mathrm{rank}(x{\restriction}_v)$.

7 Auxiliary game

Before moving towards a proof of the other implication, we need to be able to construct reasonable strategies of $\exists$ in $\mathcal{G}_\mathcal{A}$. This is achieved by considering an auxiliary game, based directly on $\mathcal{G}_\mathcal{A}$, when considering a single transition of $\mathcal{A}$ at each round. The players of the game are called Automaton (responsible for choosing transitions and choices of $\forall$ in $\mathcal{G}_\mathcal{A}$) and Pathfinder (responsible for choices of $\exists$ in $\mathcal{G}_\mathcal{A}$). The game is denoted $\mathcal{H}_{\mathcal{A},t,N}$ and depends on the fixed automaton $\mathcal{A}$, a tree $t \in \mathrm{Tr}_A$, and a number $N \in \mathbb{N}$. Positions of $\mathcal{H}_{\mathcal{A},t,N}$ are triples $(v, q, s) \in \{\mathtt{L}, \mathtt{R}\}^* \times Q \times \{\mathsf{T}, \mathsf{R}\}$, plus some additional auxiliary positions, to which we do not refer explicitly.

For a node $v \in \mathrm{dom}(t)$ and a state $q \in Q$ define

$$\mathrm{val}_t(v, q) = \inf\{\mathrm{rank}(x) \mid (t{\restriction}_v, x) \text{ can be accepted from } q\}.$$

We assume $\inf \emptyset = \infty$ (which is greater than all ordinals).

In positions (v, q, s) such that $\mathrm{val}_t(v, q) = 0$ the game reaches an *immediate victory* in which Pathfinder wins. A round from a position (v, q, s) such that $\mathrm{val}_t(v, q) > 0$ consists of the following steps:

1. Automaton declares a transition $\delta = (q, (t(v), i), q_\mathtt{L}, q_\mathtt{R})$ from the current state q over $(t(v), i)$ for some $i \in \{0, 1\}$.
2. Pathfinder declares a selector (δ, s, m, d) for (δ, s).
3. Automaton declares a direction d' that agrees with the selector.

Let $v' = vd'$, $q' = q_{d'}$, and let s' be the output side of the selector in the direction d'. Now, for every possible number $k \leq N$, the following four conditions of *immediate victory* may end the game, making Automaton win:

$s = \mathsf{T} \ \wedge \ \mathrm{val}_t(v, q) \geq \omega \cdot k \ \wedge \ s' = \mathsf{T} \wedge \mathrm{val}_t(v', q') < \omega \cdot k,$
$s = \mathsf{T} \ \wedge \ \mathrm{val}_t(v, q) \geq \omega \cdot k \ \wedge \ s' = \mathsf{R} \wedge \mathrm{val}_t(v', q') \leq \omega \cdot (k{-}1),$
$s = \mathsf{R} \ \wedge \ \mathrm{val}_t(v, q) > \omega \cdot k \ \wedge \ s' = \mathsf{R} \wedge \mathrm{val}_t(v', q') \leq \omega \cdot k,$
$s = \mathsf{R} \ \wedge \ \mathrm{val}_t(v, q) > \omega \cdot k \ \wedge \ s' = \mathsf{T} \wedge \mathrm{val}_t(v', q') < \omega \cdot k.$

If no immediate victory happened, the game proceeds to the new position which is (v', q', s'). Note that the conditions of immediate victory depend only on (v, q, s) and (v', q', s'), so can be directly hardwired in the structure of the game.

An infinite play (i.e., without any immediate victory) of $\mathcal{H}_{\mathcal{A},t,N}$ is won by Pathfinder if the sequence of visited states is rejecting or a selector with mode ② is played infinitely often.

The main result about $\mathcal{H}_{\mathcal{A},t,N}$ is the following lemma. First, the winning condition of $\mathcal{H}_{\mathcal{A},t,N}$ guarantees that if Pathfinder wins the game then he wins it positionally. To prove that he wins, one assumes that Automaton wins from some position, which leads to a contradiction.

▶ **Lemma 7.1.** *Pathfinder has a positional strategy in $\mathcal{H}_{\mathcal{A},t,N}$ that is winning from every position of the game (we call such a strategy* uniform*).*

8 Completeness

We now move to the proof of Item 2 of Proposition 4.2: if $\forall$ wins $\mathcal{G}_{\mathcal{A}}$ then $\eta_\Gamma < \omega \cdot N$ for some computable $N \in \mathbb{N}$. First observe the following important consequence of the winning condition of the game.

▶ **Proposition 8.1.** *There exists a computable bound $N \in \mathbb{N}$ such that if $\forall$ wins $\mathcal{G}_{\mathcal{A}}$ then he has a winning strategy that guarantees the following. In every play Π (either finite or infinite) consistent with this strategy, every path in the graph obtained as the composition of back-markings $\bar{\mu}_1, \bar{\mu}_2, \ldots$ contains less than N changes of sides from T to R.*

Compute the bound N as above. Assume that $\forall$ wins the game $\mathcal{G}_{\mathcal{A}}$ and fix his winning strategy satisfying the thesis of the above proposition. We claim that then the closure ordinal of the automaton is bounded by $\omega \cdot N$. Assume for the sake of contradiction that there exists a tree t which does not have a witnessing tree x of rank smaller than $\omega \cdot N$. The above assumption about t implies that $\mathrm{val}_t(\varepsilon, q_{\mathrm{I}}) \geq \omega \cdot N$.

Based on the tree t, we now construct a play that is consistent with the fixed $\forall$'s strategy, but is won be $\exists$, leading to a contradiction. Together with the current position of the game, $(\!|T_n, R_n|\!)$, we store some node v_n of the tree, starting with $v_0 = \varepsilon$. In order to construct the play, we need to provide choices of $\exists$: she declares a letter based on the label of v_n in the fixed tree t, and a set of selectors based on a fixed uniform positional strategy π_0 of Pathfinder in $\mathcal{H}_{\mathcal{A},t,N}$, given by Lemma 7.1. The current node v_n is then updated according to the direction declared by $\forall$ in $\mathcal{G}_{\mathcal{A}}$.

For a pair $(q, s) \in (\!|T_n, R_n|\!)$ denote by $\mathrm{hist}_n(q, s)$ the number of switches from the side T to the side R on the back-marked history of (q, s). We keep a *val-preserving invariant* saying that for every pair $(q, s) \in (\!|T_n, R_n|\!)$ with $k = N - \mathrm{hist}_n(q, s)$ either

1. $s = \mathsf{T}$ and $\mathrm{val}_t(v_n, q) \geq \omega \cdot k$, or
2. $s = \mathsf{R}$ and $\mathrm{val}_t(v_n, q) > \omega \cdot k$.

Note that the val-preserving invariant is initially met, because $\mathrm{val}_t(\varepsilon, q_{\mathrm{I}}) \geq \omega \cdot N$. Note also that always $k > 0$ because Proposition 8.1 implies that $\mathrm{hist}_n(q, s) < N$. The fact that the Pathfinder's strategy π_0 is winning in $\mathcal{H}_{\mathcal{A},t,N}$ can be used to deduce that the constructed play is won by $\exists$ in $\mathcal{G}_{\mathcal{A}}$. This concludes the proof of Proposition 4.2, hence also of Theorem 1.1.

9 Definability of ranks in MSO

We now move to a study of definability of particular rank bounds in MSO.

With some analogy to the cardinality quantifier by Bárány et al. [3], one can propose a quantifier $\exists^{\leq\eta} X.\, \varphi(\vec{Y}, X)$, expressing that there exists a well-founded set X of rank at most η such that $\varphi(\vec{Y}, X)$ holds. Note that this can be equivalently rewritten using the predicate $\mathrm{rank}(X) \leq \eta$, defined by $\exists^{\leq\eta} Z.\, X = Z$. Below we show that the predicate $\mathrm{rank}(X) \leq \eta$, and consequently the respective quantifier, cannot be expressed in MSO except for the basic case of natural numbers (or ω_1). Definability of this predicate in MSO boils down to checking if the following language is regular:

$$L_{\leq\eta} \stackrel{\text{def}}{=} \big\{x \in \mathrm{Tr}_{\{0,1\}} \mid \mathrm{rank}(x) \leq \eta\big\}.$$

Clearly, $\mathrm{rank}(x) < \omega_1$ holds for every well-founded tree, thus it remains to consider $\eta < \omega_1$.

We first show that these languages are regular for $\eta < \omega$.

▶ **Fact 9.1.** *For every $l < \omega$ the language $L_{\leq l}$ is regular.*

Going beyond ω, the languages stop being regular. First, a simple pumping argument shows the following.

▶ **Lemma 9.2.** *For all ordinals $\eta < \omega_1$ and all $l < \omega$ the language $L_{\leq\eta+\omega+l}$ is not regular.*

The argument from Lemma 9.2 applies in particular to all ordinals η such that $\omega \leq \eta < \omega^2$, whereas the ordinals $\eta \geq \omega^2$ are covered by Theorem 1.1. Thus we may conclude.

▶ **Corollary 9.3.** *For an ordinal $\eta < \omega_1$ the language $L_{\leq\eta}$ is regular if and only if $\eta < \omega$.*

The above corollary does not exclude the possibility that a higher ordinal η may be a supremum of a regular subset of $L_{\leq\eta}$.

▶ **Lemma 9.4.** *For each pair of natural numbers k, l there exists a regular language $L \subseteq \mathrm{WF}$ such that $\sup_{x\in L} \mathrm{rank}(x) = \omega \cdot k + l$.*

▶ Remark 9.5. Clearly no countable formalism can define all the languages $L_{\leq\eta}$ for $\eta < \omega_1$. However, certain formalisms can go beyond MSO. For instance, the logic WMSO+U (for which the satisfiability problem is known to be decidable [4]) is capable of defining the language $L_{\leq\omega}$: a tree x has rank at most ω if below every node u labelled by 1 there is a bound K on the number of nodes labelled by 1 that can appear on branches of x passing through u. This construction can be iterated to define the languages $L_{\leq\omega\cdot k+l}$ and possibly even beyond that.

10 Closure ordinals

In this section we show how a negative answer to Czarnecki's question on closure ordinals can be derived from the present result. We use the standard syntax and semantics of modal μ-calculus [7], with its formulae constructed using the following grammar:

$$F ::= a \mid X \mid \mu X.\, F \mid \nu X.\, F \mid F_1 \vee F_2 \mid F_1 \wedge F_2 \mid \Diamond F \mid \Box F,$$

where $a \in A$ is a letter from a fixed alphabet, X is a variable from some fixed set of variables, μ and ν are the least and the greatest *fixed-point operators*, and $\Diamond$ and $\Box$ are the standard modalities ("exists a successor" and "for all successors"). For technical convenience we make an assumption that, in each point of our model, exactly one proposition (letter in A) is satisfied. For the sake of readability we often identify a formula with its semantics, that is, we read a closed formula as a subset of the domain, and a formula with k free variables as a k-ary function over its powerset.

Given an ordinal number η, we use the standard notation $\mu^\eta X.\, F(X)$ for the η-approximation of the fixed point in a given model (which amounts to the η's iteration $F^\eta(\emptyset)$). Given a model τ, we define the *closure ordinal of* $\mu X.\, F(X)$ *in* τ as the least ordinal η such that $\mu X.\, F(X) = \mu^\eta X.\, F(X)$. Then, the *closure ordinal of* $\mu X.\, F(X)$ (as considered in Question 1.2) is the supremum of these ordinals over all models τ (or ∞ if the supremum does not exist). We aim at providing an (alternative to Afshari, Barlucchi, and Leigh [1]) proof of the following theorem.

▶ **Theorem 10.1.** *Let $F(X)$ be a μ-calculus formula in which the variable X does not occur in scope of any fixed-point operator. Then, the closure ordinal of $\mu X.\, F(X)$ is either strictly smaller than ω^2, or at least ω_1 and it can be effectively decided which of the cases holds.*

Note that we allow arbitrary closed formulae of μ-calculus to be nested in F; however, we do not cover the whole μ-calculus, because of the restriction on occurrences of X. This stays in line with the fragment considered by Afshari, Barlucchi, and Leigh [1] (as explained in Section 1), but we additionally provide a decision procedure that makes the dichotomy effective.

Towards a proof of Theorem 10.1, as a first step, we eliminate from F all occurrences of X that are not in scope of any modal operator; this can be done without changing the closure ordinal. Next, using standard techniques we obtain the following lemma.

▶ **Lemma 10.2.** *Let $F(X)$ be a formula as in Theorem 10.1, with all occurrences of X being in scope of a modal operator. The following three conditions are equivalent for every countable limit ordinal η:*

1. *the closure ordinal of $\mu X.\, F(X)$ is bounded by η;*
2. *for every model τ that is a countable tree with its root in $\mu X.\, F(X)$, this root belongs to $\mu^\eta X.\, F(X)$;*
3. *for every model τ that is a countable tree with its root in $\mu X.\, F(X)$, there exists a well-founded set $Z \subseteq \mathrm{dom}(\tau)$ containing the root, such that $F(Z) \supseteq Z$ and $\mathrm{rank}(Z) \leq \eta$.*

A countable tree τ, occurring in Items 2 and 3 above, can be seen as a function $\tau\colon X \to A$ from a prefix-closed subset $X \subseteq \mathbb{N}^*$ to a finite alphabet A. Now, recall a natural encoding $(n_1, n_2, \ldots, n_k) \mapsto \mathtt{R}^{n_1}\mathtt{L}\mathtt{R}^{n_2}\mathtt{L}\cdots\mathtt{R}^{n_k}\mathtt{L}$ of $\mathbb{N}^*$ into $\{\mathtt{L},\mathtt{R}\}^*$. This encoding preserves the prefix order on $\mathbb{N}^*$ and moreover preserves ranks of well-founded sets. Take the relation Γ_F that contains $(t, x) \in \mathrm{Tr}_A \times \mathrm{Tr}_{\{0,1\}}$ if:

- t encodes a model τ,
- x encodes a set $Z \subseteq \mathrm{dom}(\tau)$,
- the root of τ belongs to $\mu X.\, F(X)$,
- the root of τ belongs to Z, and $F(Z) \supseteq Z$.

The μ-calculus formulae $F(Z)$ and $\mu X.\, F(X)$ can be rewritten into MSO [7] and then modified to read the above encoding of τ in the binary tree, instead of τ itself. It follows that the relation Γ_F is MSO-definable.

Observe that Item 3 of Lemma 10.2 can be rephrased by saying that the closure ordinal of Γ_F is bounded by η. Applying Theorem 1.1 to Γ_F, and then Lemma 10.2, we have one of two possibilities: If the closure ordinal of Γ_F is smaller than ω^2, then it is bounded by $\omega \cdot N$ for some $N \in \mathbb{N}$; then also the closure ordinal of $\mu X.\, F(X)$ is bounded by $\omega \cdot N < \omega^2$.

Otherwise, the closure ordinal of Γ_F is ω_1, so it is not bounded by any countable limit ordinal; then also the closure ordinal of $\mu X. F(X)$ is not bounded by any countable limit ordinal, hence it is at least ω_1. This concludes the proof of Theorem 10.1.

▶ Remark 10.3. One may ask if Theorem 10.1 is merely a consequence or it is in some sense equivalent to our main result Theorem 1.1. To the best of our understanding, Theorem 10.1 does not transfer back to the general realm of MSO-definable relations, as in Theorem 1.1. One of the reasons is that the iterations of fixed points are required to proceed in a monotone fashion, driven by the internal formula F; while in full MSO one can express arbitrary correspondence between the parameters $\vec{Y}$ and a well-founded witness X.

11 Conclusions

This work contributes to the study of expressive power of the MSO theory of binary tree. We investigate to what extent this theory can express properties of well-founded trees, and in particular distinguish between their ordinal ranks. We observe that the ability of explicit expression of properties of ranks is practically limited to statements of the form: all trees X satisfying $\varphi(X)$ have $\mathrm{rank}(X) < N$, for a fixed $N \in \mathbb{N}$ (cf. Corollary 9.3 above). However the implicit expressive power of MSO logic goes much higher. In particular, our main result (Theorem 1.1) allows us to decide whether the property

$$\exists X. \varphi(\vec{Y}, X) \ \wedge\ X \text{ is well-founded with } \mathrm{rank}(X) < \omega^2,$$

is generally true (for all $\vec{Y}$), although the property itself is not expressible in MSO.

There is, however, a number of questions that remain to be answered. As ordinals smaller than ω^2 can be effectively represented, we would like to have an effective procedure that, given a formula φ, computes the exact bound, that is, (a representation of) the least ordinal η_φ that can be substituted for ω^2 in the construction above. Even more elementarily, given an MSO-definable set L of well-founded trees, we would like to compute the supremum of ranks of trees in L. These questions are subjects of ongoing research.

A more far-reaching direction is to relate the techniques of the present paper to the open problem of computing the Mostowski index, mentioned in Introduction. The parity condition itself imposes well-foundedness restriction on the occurrences of each odd label m in the fragments of tree where this label is the highest. Colcombet and Löding [11] have approached the index problem (still unsolved) by reducing it to the boundedness problem for distance automata (see also Idir and Lehtinen [20] for a simplified version of this reduction). One may consider an alternative approach towards the index problem by studying the ordinal ranks which arise from the well-foundedness restriction of the parity condition.

References

1 Bahareh Afshari, Giacomo Barlucchi, and Graham E. Leigh. The limit of recursion in state-based systems. In *FICS*, 2024. URL: https://www.irif.fr/_media/users/saurin/fics2024/pre-proceedings/fics-2024-afshari-etal.pdf.

2 Bahareh Afshari and Graham E. Leigh. On closure ordinals for the modal mu-calculus. In *CSL*, volume 23 of *LIPIcs*, pages 30–44. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2013. doi:10.4230/LIPICS.CSL.2013.30.

3 Vince Bárány, Łukasz Kaiser, and Alexander M. Rabinovich. Expressing cardinality quantifiers in monadic second-order logic over trees. *Fundam. Inform.*, 100(1-4):1–17, 2010. doi:10.3233/FI-2010-260.

4 Mikołaj Bojańczyk. Weak MSO+U with path quantifiers over infinite trees. In *ICALP (2)*, volume 8573 of *Lecture Notes in Computer Science*, pages 38–49. Springer, 2014. doi:10.1007/978-3-662-43951-7_4.

5 Julian C. Bradfield. Simplifying the modal mu-calculus alternation hierarchy. In *STACS*, volume 1373 of *Lecture Notes in Computer Science*, pages 39–49. Springer, 1998. doi:10.1007/BFB0028547.

6 Julian C. Bradfield, Jacques Duparc, and Sandra Quickert. Transfinite extension of the mu-calculus. In *CSL*, volume 3634 of *Lecture Notes in Computer Science*, pages 384–396. Springer, 2005. doi:10.1007/11538363_27.

7 Julian C. Bradfield and Igor Walukiewicz. The mu-calculus and model checking. In *Handbook of Model Checking*, pages 871–919. Springer, 2018. doi:10.1007/978-3-319-10575-8_26.

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

9 Lorenzo Clemente and Michał Skrzypczak. Deterministic and game separability for regular languages of infinite trees. In *ICALP*, volume 198 of *LIPIcs*, pages 126:1–126:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICS.ICALP.2021.126.

10 Thomas Colcombet, Denis Kuperberg, Christof Löding, and Michael Vanden Boom. Deciding the weak definability of Büchi definable tree languages. In *CSL*, volume 23 of *LIPIcs*, pages 215–230. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2013. doi:10.4230/LIPICS.CSL.2013.215.

11 Thomas Colcombet and Christof Löding. The non-deterministic Mostowski hierarchy and distance-parity automata. In *ICALP (2)*, volume 5126 of *Lecture Notes in Computer Science*, pages 398–409. Springer, 2008. doi:10.1007/978-3-540-70583-3_33.

12 Marek Czarnecki. How fast can the fixpoints in modal mu-calculus be reached? In *FICS*, pages 35–39. Laboratoire d'Informatique Fondamentale de Marseille, 2010. URL: https://hal.archives-ouvertes.fr/hal-00512377/document#page=36.

13 Christian Delhommé. Automaticité des ordinaux et des graphes homogènes. *Comptes Rendus de L'Académie des Sciences, Mathématiques*, 339(1):5–10, 2004.

14 Javier Esparza, Stefan Kiefer, and Michael Luttenberger. Derivation tree analysis for accelerated fixed-point computation. *Theor. Comput. Sci.*, 412(28):3226–3241, 2011. doi:10.1016/J.TCS.2011.03.020.

15 Alessandro Facchini, Filip Murlak, and Michał Skrzypczak. Rabin-Mostowski index problem: A step beyond deterministic automata. In *LICS*, pages 499–508. IEEE Computer Society, 2013. doi:10.1109/LICS.2013.56.

16 Nathanaël Fijalkow, Nathalie Bertrand, Patricia Bouyer-Decitre, Romain Brenguier, Arnaud Carayol, John Fearnley, Hugo Gimbert, Florian Horn, Rasmus Ibsen-Jensen, Nicolas Markey, Benjamin Monmege, Petr Novotný, Mickael Randour, Ocan Sankur, Sylvain Schmitz, Olivier Serre, and Mateusz Skomra. Games on graphs. *CoRR*, abs/2305.10546, 2023. doi:10.48550/arXiv.2305.10546.

17 Olivier Finkel and Stevo Todorcevic. Automatic ordinals. *Int. J. Unconv. Comput.*, 9(1-2):61–70, 2013. URL: http://www.oldcitypublishing.com/journals/ijuc-home/ijuc-issue-contents/ijuc-volume-9-number-1-2-2013/ijuc-9-1-2-p-61-70/.

18 Gaëlle Fontaine. Continuous fragment of the mu-calculus. In *CSL*, volume 5213 of *Lecture Notes in Computer Science*, pages 139–153. Springer, 2008. doi:10.1007/978-3-540-87531-4_12.

19 Maria J. Gouveia and Luigi Santocanale. $\aleph_1$ and the modal μ-calculus. *Log. Methods Comput. Sci.*, 15(4), 2019. doi:10.23638/LMCS-15(4:1)2019.

20 Olivier Idir and Karoliina Lehtinen. Mostowski index via extended register games. *CoRR*, abs/2412.16793, 2024. doi:10.48550/arXiv.2412.16793.

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

22 Damian Niwiński. On the cardinality of sets of infinite trees recognizable by finite automata. In *MFCS*, volume 520 of *Lecture Notes in Computer Science*, pages 367–376. Springer, 1991. doi:10.1007/3-540-54345-7_80.

23 Damian Niwiński and Igor Walukiewicz. A gap property of deterministic tree languages. *Theor. Comput. Sci.*, 1(303):215–231, 2003. doi:10.1016/S0304-3975(02)00452-8.

24 C.-H. Luke Ong. On model-checking trees generated by higher-order recursion schemes. In *LICS*, pages 81–90. IEEE Computer Society, 2006. doi:10.1109/LICS.2006.38.

25 Michael O. Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of the American Mathematical Society*, 141:1–35, 1969.

26 Sylvain Schmitz. *Algorithmic Complexity of Well-Quasi-Orders. (Complexité algorithmique des beaux pré-ordres)*. École normale supérieure Paris-Saclay, 2017. URL: https://tel.archives-ouvertes.fr/tel-01663266.

27 Michał Skrzypczak. *Descriptive Set Theoretic Methods in Automata Theory – Decidability and Topological Complexity*, volume 9802 of *Lecture Notes in Computer Science*. Springer, 2016. doi:10.1007/978-3-662-52947-8.

28 Michał Skrzypczak and Igor Walukiewicz. Deciding the topological complexity of Büchi languages. In *ICALP*, volume 55 of *LIPIcs*, pages 99:1–99:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPICS.ICALP.2016.99.

29 Wolfgang Thomas. Languages, automata, and logic. In *Handbook of Formal Languages (3)*, pages 389–455. Springer, 1997. doi:10.1007/978-3-642-59126-6_7.