On Guidable Index of Tree Automata

Damian Niwiński
Institute of Informatics, University of Warsaw, Poland

Michał Skrzypczak
Institute of Informatics, University of Warsaw, Poland

Abstract

We study guidable parity automata over infinite trees introduced by Colcombet and Löding, which form an expressively complete subclass of all non-deterministic tree automata. We show that, for any non-deterministic automaton, an equivalent guidable automaton with the smallest possible index can be effectively found. Moreover, if an input automaton is of a special kind, i.e. it is deterministic or game automaton then a guidable automaton with an optimal index can be deterministic (respectively game) automaton as well. Recall that the problem whether an equivalent non-deterministic automaton with the smallest possible index can be effectively found is open, and a positive answer is known only in the case when an input automaton is a deterministic, or more generally, a game automaton.

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

Keywords and phrases guidable automata, index problem, ω-regular games

Digital Object Identifier 10.4230/LIPIcs.MFCS.2021.81

Funding *Michał Skrzypczak*: Supported by the Polish NCN grant 2016/22/E/ST6/00041.

1 Introduction

Rabin automata on infinite trees constitute one of the most expressive formalisms based on finite-state recognisability. Introduced by Rabin in the proof of his seminal decidability result [19] as an enhancement of the more basic concept of Büchi automata on ω-words [2], tree automata continue to attract attention of researchers, and make a crossing point of various areas like logic, games, set-theoretic topology, fixed-point calculi, and theory of verification. It is an intriguing fact that a number of questions, which are by now well understood for automata on ω-words, are still largely unsolved for automata on trees. This concerns in particular decidability questions like effective measurability and effective classification within various hierarchies based on topology, or on the structure of automata. While it is plausible that we reach the frontiers of decidability here, there is neither any evidence that these problems may not be decidable.

Among the hierarchies mentioned above, the Rabin-Mostowski index hierarchy has attracted a special attention, owing to its close relation to an alternation hierarchy in the μ-calculus (cf. [1, 17]), and to the complexity of the non-emptiness problem. It refers to the acceptance criterion, which for infinite computations is conveniently expressed in terms of colours associated with the automaton states, that may or should repeat infinitely often. In a general setting, the criterion specifies explicitly the requested combinations of colours (Muller criterion), which refines the original Büchi criterion that has just requested some "good" colour to repeat infinitely often. While the Büchi and Muller criteria are equivalent for non-deterministic automata on ω-words [2], they are not for automata on trees as noted already by Rabin [20]. The parity criterion is derived from Büchi's idea by distinguishing between "good" and "bad" colours and requesting that the highest-ranked colour repeating infinitely often is good. This is implemented by representing good and bad colours by even and odd integers, respectively. An *index* is a set of colours $\{i, i+1, \ldots, j\}$ used by an automaton; its size corresponds to the number of alternations between good and bad

46th International Symposium on Mathematical Foundations of Computer Science (MFCS 2021).
Editors: Filippo Bonchi and Simon J. Puglisi; Article No. 81; pp. 81:1–81:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

colours. The parity criterion is expressively equivalent to the Muller criterion, but it is more succinct and leads to the better complexity of the non-emptiness problem, up to our present knowledge. (See [21] for an insightful discussion of other meaningful criteria.)

The complexity of the non-emptiness problem for automata on infinite trees is actually a pertinent open question. While the problem is NP-complete for automata with the original Rabin acceptance criterion [9], it is expectedly more feasible for automata with the parity criterion, being in turn equivalent to the celebrated problem of solving parity games [10]. It is well-known that the complexity of this last problem has been recently improved [4] roughly from $n^{\mathcal{O}(d)}$ to $n^{\mathcal{O}(\log d)}$, where n is the number of positions and d the number of priorities (colours). This yields a quasi-polynomial solution for the non-emptiness problem for parity tree automata as well, with d corresponding to the size of automaton's index. Whether this improvement makes the quest for a small index more or less important may be the subject of discussion. In prospective applications in the model checking, the index d is derived from the formula whereas n corresponds to the size of the system, and is usually much bigger than d, so that even a small improvement of the index may be advantageous. But in any case, the recent development for parity games shows that the index is an essential parameter in understanding the problem.

The relevance of this parameter has been noticed already by Mostowski [15]. While it is known that in general an arbitrarily high index is needed for non-deterministic [16] as well as for alternating [1] automata, it is generally open if an automaton with an optimal index can be effectively found. Algorithms were given only in the special cases, where the input automaton is deterministic [18], or more generally it is a so-called game automaton [11].

An interesting approach has been undertaken by Colcombet and Löding [7], who reduced the non-deterministic index problem of parity tree automata to a problem of an apparently different nature, concerning the asymptotic behaviour of counters in some automata related to the (classical) star-height problem. In the course of their proof, these authors introduced an auxiliary concept of a *guidable tree automaton*, which in our opinion has a compelling potential and deserves to be better understood. Intuitively, a guidable automaton behaves almost like a deterministic automaton if it is given (as a kind of oracle) the non-deterministic choices of *any* other automaton running on the same tree. Colcombet and Löding [7] showed in particular that any automaton can be transformed into an equivalent guidable automaton, which essentially realises a strategy of Pathfinder in the celebrated game introduced by Gurevich and Harrington [13]. Although a guidable automaton is unsurprisingly not unique, it nevertheless constitutes a kind of a normal form of a non-deterministic automaton. It should be stressed that canonical forms are generally missing in the theory of automata on infinite objects, which is one of the sources of difficulties there. Therefore, we believe that the idea of guidable automata is worth to pursue.

In the present paper we show how to effectively compute an equivalent guidable automaton with the smallest possible index – among all guidable automata – without any restriction on the input automaton. We also revisit the concept of game automata mentioned above and show, relying on the construction of a guidable automaton by Colcombet and Löding [7], that any game automaton is itself guidable. Moreover, if an input automaton is a game or deterministic automaton then a guidable automaton with an optimal index can be a game (respectively, deterministic) automaton as well.

Unfortunately, the guidable index can be in general arbitrarily worse than the smallest possible index of an equivalent non-deterministic automaton, we are primarily searching for. We believe however, that the games used in our proofs may be of potential interest for the main problem. Note that similar games have been used in a recent work [5] (see [6]) to decide separability of regular tree languages by deterministic and game automata.

2 Basic notions

The set of natural numbers $\{0, 1, \ldots\}$ is denoted ω. An *alphabet* is any finite non-empty set Σ of *letters*. By Σ^* we denote the set of *words* (i.e. finite sequences) over an alphabet Σ. By $u{\restriction}_n$ we denote the finite sequence consisting of the first n symbols of u. The empty sequence is denoted ϵ. By $|u|$ we denote the length of a finite sequence u.

A Σ-*labelled tree* (shortly *tree*) is a function $t\colon \{\mathtt{L}, \mathtt{R}\}^* \to \Sigma$, where $\mathtt{L} \neq \mathtt{R}$ are two special symbols called *directions*. The set of all such trees is denoted Tr_Σ. If Σ and Γ are two alphabets then each pair of trees $t_1 \in \mathrm{Tr}_\Sigma$ and $t_2 \in \mathrm{Tr}_\Gamma$ induces their *product* $t_1 \otimes t_2 \in \mathrm{Tr}_{\Sigma\times\Gamma}$, with $(t_1 \otimes t_2)(u) \stackrel{\text{def}}{=} \big(t_1(u), t_2(u)\big)$. Given a tree language $L \subseteq \mathrm{Tr}_{\Sigma\times\Gamma}$, its *projection* onto Σ is the set of trees $t_1 \in \mathrm{Tr}_\Sigma$, such that there exists a tree $t_2 \in \mathrm{Tr}_\Gamma$, for which $t_1 \otimes t_2 \in L$.

An infinite sequence of directions $b \in \{\mathtt{L}, \mathtt{R}\}^\omega$ is called a *branch*. Similarly, a *path* in a tree $t \in \mathrm{Tr}_A$ is a sequence $(a_n, d_n)_{n\in\omega} \in \big(\Sigma \times \{\mathtt{L}, \mathtt{R}\}\big)^\omega$, such that $a_n = t(d_0 \cdots d_{n-1})$ for $n = 0, 1, \ldots$

An *index* is a non-empty finite range of natural numbers $C = \{i, i{+}1, \ldots, j\} \subseteq \omega$. Elements $c \in C$ are called *priorities*. We say that an infinite sequence of priorities $(c_n)_{n\in\omega}$ is *parity accepting* if $\limsup_{n\to\infty} c_n \equiv 0 \mod 2$. Otherwise, we say that such a sequence is *parity rejecting*.

A *non-deterministic C-parity tree automaton* (shortly *non-deterministic automaton*) is a tuple $\mathcal{A} = \langle \Sigma, Q^\mathcal{A}, q_\mathrm{I}^\mathcal{A}, \Delta^\mathcal{A}, \Omega^\mathcal{A} \rangle$, where Σ is an alphabet, $Q^\mathcal{A}$ a finite set of *states*, $q_\mathrm{I}^\mathcal{A} \in Q^\mathcal{A}$ an *initial state*, $\Delta^\mathcal{A} \subseteq Q^\mathcal{A} \times \Sigma \times Q^\mathcal{A} \times Q^\mathcal{A}$ a *transition relation*; and $\Omega^\mathcal{A}\colon Q^\mathcal{A} \to C$ a *priority mapping*. An element $(q, a, q_\mathtt{L}, q_\mathtt{R}) \in \Delta^\mathcal{A}$ is called a *transition* of the automaton $\mathcal{A}$. We say that such a transition is *from* the state q and is *over* the letter a. We make a proviso that, unless stated otherwise, all automata in consideration are *trimmed*, that is, for each state $q \in Q^\mathcal{A}$ and letter $a \in \Sigma$, there is at least one transition from q over a in $\Delta^\mathcal{A}$. When an automaton $\mathcal{A}$ is known from the context then we skip the superscript and write just Q, Δ, etc.

We extend the notions of parity accepting (resp. rejecting) sequences of priorities to sequences of states by applying Ω, i.e. a sequence of states $(q_n)_{n\in\omega}$ is *parity accepting* (resp. *parity rejecting*) in $\mathcal{A}$ if the priorities $(\Omega(q_n))_{n\in\omega}$ are parity accepting (resp. parity rejecting).

Given a tree $t \in \mathrm{Tr}_\Sigma$, a *run* of an automaton $\mathcal{A}$ over t is a tree $\rho \in \mathrm{Tr}_Q$ such that $\rho(\epsilon) = q_\mathrm{I}$ and, for each node $u \in \{\mathtt{L}, \mathtt{R}\}^*$, the tuple

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

is a transition of $\mathcal{A}$. Such a run is *accepting* if, for every branch $b \in \{\mathtt{L}, \mathtt{R}\}^\omega$, the sequence of states $q_n \stackrel{\text{def}}{=} \rho\big(b{\restriction}_n\big)$ for $n = 0, 1, \ldots$ is *parity accepting*. The set of trees over which a given automaton $\mathcal{A}$ has some accepting run is called the *language* of $\mathcal{A}$ and is denoted $\mathrm{L}(\mathcal{A})$. If $q \in Q$ is a state of an automaton $\mathcal{A}$ then by $\mathrm{L}(\mathcal{A}, q)$ we denote the language of the automaton $\mathcal{A}$ with the initial state set to q. We say that q is *productive* if $\mathrm{L}(\mathcal{A}, q) \neq \emptyset$. Thus $\mathrm{L}(\mathcal{A}) \neq \emptyset$ iff the initial state $q_\mathrm{I}^\mathcal{A}$ is productive.

A set of trees is a *regular tree language* if it is of the form $\mathrm{L}(\mathcal{A})$ for some non-deterministic automaton $\mathcal{A}$.

An automaton $\mathcal{A}$ is *deterministic* if for every state $q \in Q$ and letter $a \in \Sigma$ there exists a unique transition in $\Delta^\mathcal{A}$ starting from q over a. A deterministic automaton has exactly one run over each tree $t \in \mathrm{Tr}_\Sigma$ and this run can be constructed inductively in a top-down way.

Guidable automata

The notion of a guiding function and one automaton guiding another was introduced in [7], here we follow the exposition from [14].

Fix two non-deterministic automata $\mathcal{A}$ and $\mathcal{B}$ over the same alphabet Σ. A *guiding function* from $\mathcal{B}$ to $\mathcal{A}$ is a function $g\colon Q^{\mathcal{A}} \times \Delta^{\mathcal{B}} \to \Delta^{\mathcal{A}}$ such that $g\big(p,(q,a,q_{\mathtt{L}},q_{\mathtt{R}})\big) = (p,a,p_{\mathtt{L}},p_{\mathtt{R}})$ for some $p_{\mathtt{L}},\ p_{\mathtt{R}} \in Q^{\mathcal{A}}$ (i.e. the function g is compatible with the state p and the letter a). If $\rho \in \mathrm{Tr}_{Q^{\mathcal{B}}}$ is a run of $\mathcal{B}$ over a tree $t \in \mathrm{Tr}_{\Sigma}$ then we define the tree $\vec{g}(\rho) \in \mathrm{Tr}_{Q^{\mathcal{A}}}$, inductively as follows. We let $\vec{g}(\rho)(\epsilon) \stackrel{\text{def}}{=} q_{\mathrm{I}}^{\mathcal{A}}$ and, for each $u \in \{\mathtt{L},\mathtt{R}\}^*$, if $\vec{g}(\rho)(u) = p \in Q^{\mathcal{A}}$, $t(u) = a$, and γ is a transition of $\mathcal{B}$ taken in u, i.e. $\gamma \stackrel{\text{def}}{=} \big(\rho(u), a, \rho(u\mathtt{L}), \rho(u\mathtt{R})\big)$, and if finally $g(p,\gamma) = (p,a,p_{\mathtt{L}},p_{\mathtt{R}}) \in \Delta^{\mathcal{A}}$, then we let,

$$\vec{g}(\rho)(u\mathtt{L}) \stackrel{\text{def}}{=} p_{\mathtt{L}}, \quad \vec{g}(\rho)(u\mathtt{R}) \stackrel{\text{def}}{=} p_{\mathtt{R}}.$$

Notice that directly by the definition, the tree $\vec{g}(\rho)$ is a run of $\mathcal{A}$ over t. We say that a guiding function $g\colon Q^{\mathcal{A}} \times \Delta^{\mathcal{B}} \to \Delta^{\mathcal{A}}$ *preserves acceptance* if whenever ρ is an accepting run of $\mathcal{B}$ then $\vec{g}(\rho)$ is an accepting run of $\mathcal{A}$. We say that an automaton $\mathcal{B}$ *guides* an automaton $\mathcal{A}$ (denoted $\mathcal{B} \hookrightarrow \mathcal{A}$), if there exists a guiding function $g\colon Q^{\mathcal{A}} \times \Delta^{\mathcal{B}} \to \Delta^{\mathcal{A}}$ which preserves acceptance. In particular, it implies that $\mathrm{L}(\mathcal{B}) \subseteq \mathrm{L}(\mathcal{A})$.

An automaton $\mathcal{A}$ is *guidable* if it can be guided by any automaton $\mathcal{B}$ such that $\mathrm{L}(\mathcal{B}) = \mathrm{L}(\mathcal{A})$ (in fact one can equivalently require that $\mathrm{L}(\mathcal{B}) \subseteq \mathrm{L}(\mathcal{A})$, see [14, Remark 4.5]).

The main result concerning guidable automata is the following theorem.

▶ **Theorem 1** ([7, Theorem 1], see also [14, Theorem 4.7]). *For every regular tree language L there exists a guidable automaton recognising L. Moreover, such an automaton can be effectively constructed from any non-deterministic automaton for L.*

Since we will rely on the exact structure of such an automaton, we recall the crucial steps in the construction. First, let $\mathcal{A}$ be a non-deterministic automaton recognising the given language L. Assume that $\mathcal{A}' = \langle \Sigma, Q', q'_{\mathrm{I}}, \Delta', \Omega' \rangle$ is any automaton recognising the complement of L.

An important role in this construction is played by *selectors*, i.e. functions $f\colon \Delta' \to \{\mathtt{L},\mathtt{R}\}$ (in other words $f \in \{\mathtt{L},\mathtt{R}\}^{\Delta'}$). Consider a product tree $t \otimes \tau \in \mathrm{Tr}_{\Sigma \times \{\mathtt{L},\mathtt{R}\}^{\Delta'}}$ and one of its paths $\big((a_n, f_n), d_n\big)_{n\in\omega}$. We say that this path is *losing* if there exists a sequence of transitions $\big(\delta'_n = (p_n, a_n, p_{\mathtt{L},n}, p_{\mathtt{R},n})\big)_{n\in\omega}$ of $\mathcal{A}'$ which is $\mathcal{A}'$*-accepting* in the following sense:

- $p_0 = q_{\mathrm{I}}^{\mathcal{A}'}$,
- $\forall n \in \omega.\ p_{n+1} = p_{d_n,n} \land d_n = f_n(\delta'_n)$,
- the sequence of states $(p_n)_{n\in\omega}$ is parity accepting in $\mathcal{A}'$.

Lemma 1.15 in [14] provides a deterministic automaton $\mathrm{Winning}(\mathcal{A}')$ over the alphabet $\Sigma \times \{\mathtt{L},\mathtt{R}\}^{\Delta'}$. The crucial properties of this automaton are stated in the following fact.

▶ **Fact 2.** *The automaton $\mathrm{Winning}(\mathcal{A}')$ accepts a product tree $t \otimes \tau \in \mathrm{Tr}_{\Sigma \times \{\mathtt{L},\mathtt{R}\}^{\Delta'}}$ if and only if none of its paths is losing. The projection of $\mathrm{L}\big(\mathrm{Winning}(\mathcal{A}')\big)$ onto Σ is the original language L, i.e. the complement of $\mathrm{L}(\mathcal{A}')$.*

By $\mathrm{Complement}(\mathcal{A}')$ we denote the non-deterministic automaton obtained from the automaton $\mathrm{Winning}(\mathcal{A}')$ by projecting the transitions of $\mathrm{Winning}(\mathcal{A}')$ onto the coordinate Σ: each transition of the form $\big(q,(a,f),q_{\mathtt{L}},q_{\mathtt{R}}\big)$ is replaced by $\big(q,a,q_{\mathtt{L}},q_{\mathtt{R}}\big)$; yielding a non-deterministic automaton recognising L. We say that a transition $\big(q,a,q_{\mathtt{L}},q_{\mathtt{R}}\big)$ of $\mathrm{Complement}(\mathcal{A}')$ is *arising* from $f \in \{\mathtt{L},\mathtt{R}\}^{\Delta'}$ if $\big(q,(a,f),q_{\mathtt{L}},q_{\mathtt{R}}\big)$ is a transition of $\mathrm{Winning}(\mathcal{A}')$.

Notice that in fact the only non-determinism of the automaton Complement($\mathcal{A}'$) comes from the projection from the alphabet $\Sigma \times \{\mathtt{L},\mathtt{R}\}^{\Delta'}$ onto Σ. Therefore, to define a guiding function to guide the automaton Complement($\mathcal{A}'$), whenever specifying a transition of Complement($\mathcal{A}'$), it is enough to provide a selector $f \in \{\mathtt{L},\mathtt{R}\}^{\Delta'}$ and then take the unique transition arising from f.

▶ **Lemma 3** (See the proof of [14, Theorem 4.7]). *The automaton* Complement($\mathcal{A}'$) *is guidable.*

3 Guidability relation

The guidability relation $\mathcal{B} \hookrightarrow \mathcal{A}$ can be seen as a reduction, showing that one automaton uses *less non-determinism* than the other one. Thus, one would naturally expect this relation to be transitive, which is indeed the case.

▶ **Proposition 4.** *If $\mathcal{C} \hookrightarrow \mathcal{B}$ and $\mathcal{B} \hookrightarrow \mathcal{A}$ then $\mathcal{C} \hookrightarrow \mathcal{A}$.*

A proof of this fact is implicit in [14, Proposition 4.11], where it is shown how to compose two guiding functions $g^{\mathcal{B}}$ and $g^{\mathcal{A}}$ into a winning strategy in a game characterising guidability, called *weak inclusion game*, see [14, page 78]. However, the transitivity of the relation $\hookrightarrow$ has not been explicitly stated before. Therefore, for the sake of completeness, we provide a proof of Proposition 4 here.

First, we recall the notion of the *weak inclusion game* $\mathcal{G}_{\mathrm{guide}}(\mathcal{B},\mathcal{A})$ from [14, page 78] which is used to characterise when an automaton $\mathcal{B}$ guides another automaton $\mathcal{A}$, and consequently to decide whether a non-deterministic automaton is guidable. We introduce a small modification, justified below, using the following concept: a transition $(q, a, q_{\mathtt{L}}, q_{\mathtt{R}})$ is *productive* if both states $q_{\mathtt{L}}, q_{\mathtt{R}}$ are productive (cf. Section 2).

The positions of the game $\mathcal{G}_{\mathrm{guide}}(\mathcal{B},\mathcal{A})$ are of the form $(p, q) \in Q^{\mathcal{A}} \times Q^{\mathcal{B}}$. The initial position (p_0, q_0) is $(q_{\mathrm{I}}^{\mathcal{A}}, q_{\mathrm{I}}^{\mathcal{B}})$. At an nth round for $n = 0, 1, \ldots$ which starts in a position (p_n, q_n):

1. $\forall$ chooses a letter $a_n \in \Sigma$,
2. $\forall$ chooses a productive transition $\gamma_n = (q_n, a_n, q_{\mathtt{L},n}, q_{\mathtt{R},n}) \in \Delta^{\mathcal{B}}$; if there is no productive transition from q_n over a_n then $\forall$ loses immediately,
3. $\exists$ chooses a transition $\delta_n = (p_n, a_n, p_{\mathtt{L},n}, p_{\mathtt{R},n}) \in \Delta^{\mathcal{A}}$,
4. $\forall$ chooses a direction $d_n \in \{\mathtt{L},\mathtt{R}\}$.

The next position (p_{n+1}, q_{n+1}) of the game is $(p_{d_n,n}, q_{d_n,n})$.

The winning condition for $\exists$ expresses, that if the sequence of states $(q_n)_{n\in\omega}$ is parity accepting in $\mathcal{B}$ then the sequence of states $(p_n)_{n\in\omega}$ is parity accepting in $\mathcal{A}$.

▶ **Lemma 5** ([14, Proposition 4.9]). *The player $\exists$ wins the above game $\mathcal{G}_{\mathrm{guide}}(\mathcal{B},\mathcal{A})$ if and only if $\mathcal{B} \hookrightarrow \mathcal{A}$. Moreover, the player $\exists$ has a positional winning strategy.*

The requirement that the transitions chosen by $\forall$ in step 2 are productive is added because of the case when $\mathrm{L}(\mathcal{B}) = \emptyset$. In this corner case the relation $\mathcal{B} \hookrightarrow \mathcal{A}$ holds trivially whereas the characterisation as stated in [14, Proposition 4.9] seems to fail. For the sake of completeness we recall the proof of Proposition 4.9 from [14], taking into account our modification.

Proof. The fact that the game is determined and the player $\exists$ has a positional winning strategy follows from the general theory of infinite games (see, e.g. [21, 12]); indeed the winning criterion of $\mathcal{G}_{\mathrm{guide}}(\mathcal{B},\mathcal{A})$ can be easily presented as a union of parity criteria and hence it is a so-called Rabin criterion. Clearly, the fact that $\exists$ can also win in finite time does not affect the positional determinacy for this player.

Now if $\mathcal{B}$ guides $\mathcal{A}$ then an acceptance-preserving guiding function $g\colon Q^{\mathcal{A}} \times \Delta^{\mathcal{B}} \to \Delta^{\mathcal{A}}$ yields a positional strategy for $\exists$: in a position (p_n, q_n), given the choice by $\forall$ of a transition $\gamma_n = (q_n, a_n, q_{\mathtt{L},n}, q_{\mathtt{R},n}) \in \Delta^{\mathcal{B}}$, $\exists$ chooses $\delta_n = g(p_n, \gamma_n)$. Conversely, a positional winning strategy of $\exists$ induces a partial function $\tilde{g}\colon Q^{\mathcal{A}} \times \Delta^{\mathcal{B}} \to \Delta^{\mathcal{A}}$ that is defined for $p \in Q^{\mathcal{A}}$ and $\gamma = (q, a, q_{\mathtt{L}}, q_{\mathtt{R}}) \in \Delta^{\mathcal{B}}$ whenever the position (p, q) is winning for $\exists$ and the transition γ is productive. We extend $\tilde{g}$ to a total guiding function $g\colon Q^{\mathcal{A}} \times \Delta^{\mathcal{B}} \to \Delta^{\mathcal{A}}$ that for all other arguments is defined in any way; note that it is always possible since, by our proviso, the automaton $\mathcal{A}$ is trimmed. By the assumption, the initial position $(q_{\mathrm{I}}^{\mathcal{A}}, q_{\mathrm{I}}^{\mathcal{B}})$ is winning. Then the fact that $\tilde{g}$ has been derived from a winning strategy implies that the function g preserves acceptance. ◀

▶ Remark 6. Based on the above lemma, the notion that a guiding function *preserves acceptance* can be equivalently rephrased as follows. Consider any sequence of productive transitions $\big(\gamma_n = (q_n, a_n, q_{\mathtt{L},n}, q_{\mathtt{R},n})\big)_{n\in\omega}$ of $\mathcal{B}$ and a sequence of directions $(d_n)_{n\in\omega}$. Assume that $q_0 = q_{\mathrm{I}}^{\mathcal{B}}$ and $q_{n+1} = q_{d_n,n}$ for $n = 0, 1, \ldots$ Let $\delta_n = (p_n, a_n, p_{\mathtt{L},n}, p_{\mathtt{R},n})$ be defined inductively for $n = 0, 1, \ldots$ by $p_0 \stackrel{\text{def}}{=} q_{\mathrm{I}}^{\mathcal{A}}$, $\delta_n = g(p_n, \gamma_n)$, and $p_{n+1} = p_{d_n,n}$. Then the condition that g *preserves acceptance* boils down to saying that if $(q_n)_{n\in\omega}$ is parity accepting in $\mathcal{B}$ then $(p_n)_{n\in\omega}$ is parity accepting in $\mathcal{A}$.

We can now move to a proof of Proposition 4. Assume that $g^{\mathcal{B}}$ and $g^{\mathcal{A}}$ are two guiding functions which preserve acceptance, witnessing that $\mathcal{C} \hookrightarrow \mathcal{B}$ and $\mathcal{B} \hookrightarrow \mathcal{A}$.

We will show that $\exists$ has a winning strategy in $\mathcal{G}_{\text{guide}}(\mathcal{C}, \mathcal{A})$, using the set of states of $\mathcal{B}$ as a memory structure. More precisely, $\exists$ keeps track of an additional state $r_n \in Q^{\mathcal{B}}$ with $r_0 \stackrel{\text{def}}{=} q_{\mathrm{I}}^{\mathcal{B}}$. Consider an nth round starting in a position (p_n, q_n) with a memory state r_n of $\exists$. Assume that $\forall$ has played $a_n \in \Sigma$ and $\gamma_n \in \Delta^{\mathcal{C}}$ as above. Let $\gamma'_n = (r_n, a_n, r_{\mathtt{L},n}, r_{\mathtt{R},n}) \stackrel{\text{def}}{=} g^{\mathcal{B}}(r_n, \gamma_n) \in \Delta^{\mathcal{B}}$ be the transition of $\mathcal{B}$ given by $g^{\mathcal{B}}$. Note that if the transition γ_n was productive then γ'_n must be productive as well. Similarly, let $\delta_n \stackrel{\text{def}}{=} g^{\mathcal{A}}(p_n, \gamma'_n)$ be the transition of $\mathcal{A}$ given by $g^{\mathcal{A}}$. Let $\exists$ play δ_n as her choice in that round. Once the round is finished, let the new memory state of $\exists$ be $r_{n+1} \stackrel{\text{def}}{=} r_{d_n,n}$.

▷ Claim 7. The strategy defined above is in fact winning for $\exists$.

Proof. Consider a play of $\mathcal{G}_{\text{guide}}(\mathcal{C}, \mathcal{A})$ that was played according to the above strategy. Let $(r_n)_{n\in\omega}$ be the sequence of memory states of $\exists$ used during that play. Assume that $(q_n)_{n\in\omega}$ is parity accepting in $\mathcal{C}$. Since $g^{\mathcal{B}}$ represents a winning strategy of $\exists$ in $\mathcal{G}_{\text{guide}}(\mathcal{C}, \mathcal{B})$, we know that $(r_n)_{n\in\omega}$ is parity accepting in $\mathcal{B}$. Therefore, we can use the fact that $g^{\mathcal{A}}$ represents a winning strategy of $\exists$ in $\mathcal{G}_{\text{guide}}(\mathcal{B}, \mathcal{A})$ and entail that $(p_n)_{n\in\omega}$ is parity accepting in $\mathcal{A}$. ◁

This concludes the proof of Proposition 4.

▶ **Corollary 8** (See [14, Proposition 4.11]). *Consider an automaton $\mathcal{A}$ and a guidable automaton $\mathcal{B}$, both recognising the same language L. Then the automaton $\mathcal{A}$ is guidable if and only if $\mathcal{B} \hookrightarrow \mathcal{A}$.*

Note that the above corollary combined with Lemma 5 and Theorem 1 yield a procedure to decide whether a non-deterministic automaton is guidable ([14, Theorem 4.7]).

4 Game automata as guidable automata

It is straightforward to see that deterministic automata are guidable (the function g in the definition above does not depend on the argument in $\Delta^{\mathcal{B}}$). Hence, guidable automata can be viewed as a semantic extension of deterministic automata. However, there already exists an established notion of automata naturally extending the deterministic ones, namely the *game automata*, see [8]. In this section we recall this notion in the framework of non-deterministic automata, and prove the following new result.

▶ **Proposition 9.** *Every game automaton is guidable.*

Since every deterministic automaton is also game, we get a syntactic stratification:

$$\text{Deterministic} \subsetneq \text{Game} \subsetneq \text{Guidable}, \tag{1}$$

with each consecutive class not only containing more automata but also recognising more languages. Theorem 1 means that the last class in this stratification is expressively complete for all regular tree languages.

We say that a non-deterministic tree automaton $\mathcal{A}$ is a *game automaton* if it satisfies the following two properties. First of all, Q contains a non-initial all-accepting state $\top \in Q$: $\Omega(\top)$ is even and for each letter $a \in A$ there is a unique transition starting from $\top$ over a, namely $(\top, a, \top, \top)$ (thus $\mathrm{L}(\mathcal{A}, \top) = \mathrm{Tr}_\Sigma$). Moreover, for a state $q \in Q - \{\top\}$ and a letter $a \in \Sigma$ we say that (q, a) is in one of the following *modes*:

- **conjunctive**: there is a unique transition starting from q over a of the form $(q, a, q_\mathtt{L}, q_\mathtt{R})$ with both $q_\mathtt{L}$ and $q_\mathtt{R}$ distinct from $\top$;
- **disjunctive**: there are exactly two transitions starting from q over a, one of the form $(q, a, q_\mathtt{L}, \top)$ and the other of the form $(q, a, \top, q_\mathtt{R})$, with both $q_\mathtt{L}$ and $q_\mathtt{R}$ distinct from $\top$.

The above definition is a direct translation of the alternating formulation of game automata into the format of non-deterministic automata. The two modes above correspond to the use of $\wedge$ or $\vee$ in the transition formula $\delta(q, a)$, see [11, Definition 3.2].

Before we move to a proof of Proposition 9, we need to recall that game automata admit a syntactic complementation construction. Fix a game automaton $\mathcal{A} = \langle \Sigma, Q, q_\mathrm{I}, \Delta, \Omega \rangle$. Consider another game automaton denoted $\mathcal{A}^\mathrm{c}$ defined as $\mathcal{A}^\mathrm{c} = \langle \Sigma, Q, q_\mathrm{I}, \Delta', \Omega' \rangle$, where Δ' and Ω' are defined as follows. First of all, $\Omega'(q) \stackrel{\text{def}}{=} \Omega(q) + 1$ for all q distinct than $\top$ and $\Omega'(\top) \stackrel{\text{def}}{=} \Omega(\top)$. Moreover, Δ' contains the following transitions:

- for each $a \in \Sigma$ we have $(\top, a, \top, \top) \in \Delta'$;
- if $q \in Q - \{\top\}$ and $a \in \Sigma$ are in conjunctive mode and $(q, a, q_\mathtt{L}, q_\mathtt{R}) \in \Delta$ then the mode of (q, a) in $\mathcal{A}^\mathrm{c}$ is disjunctive, with $(q, a, q_\mathtt{L}, \top)$ and $(q, a, \top, q_\mathtt{R})$ in Δ';
- if $q \in Q - \{\top\}$ and $a \in \Sigma$ are in disjunctive mode and $(q, a, q_\mathtt{L}, \top)$, $(q, a, \top, q_\mathtt{R}) \in \Delta$ then the mode of (q, a) in $\mathcal{A}^\mathrm{c}$ is conjunctive, with $(q, a, q_\mathtt{L}, q_\mathtt{R}) \in \Delta'$.

The following fact is an immediate consequence of the alternating semantics of game automata, see [11, page 24:7].

▶ **Fact 10.** *The automaton* $\mathcal{A}^\mathrm{c}$ *is also a game automaton and* $\mathrm{L}(\mathcal{A}^\mathrm{c}) = \mathrm{Tr}_\Sigma - \mathrm{L}(\mathcal{A})$.

The rest of this section is devoted to a proof of Proposition 9. Fix a game automaton $\mathcal{A} = \langle \Sigma, Q, q_\mathrm{I}, \Delta, \Omega \rangle$ and let $\mathcal{A}' \stackrel{\text{def}}{=} \mathcal{A}^\mathrm{c}$ denote its syntactic complement defined as above. Let Δ' be the set of transitions of $\mathcal{A}'$. Notice that (up to subtracting 2 from Ω) the automaton $(\mathcal{A}')^\mathrm{c}$ is equal to the original automaton $\mathcal{A}$. We will now apply the procedure of constructing a guidable automaton for L. Recall that this procedure involves a deterministic automaton $\mathrm{Winning}(\mathcal{A}')$ that accepts a product tree $t \otimes \tau \in \mathrm{Tr}_{\Sigma \times \{\mathtt{L},\mathtt{R}\}^{\Delta'}}$ if and only if none of its paths is losing. The desired guidable automaton $\mathrm{Complement}(\mathcal{A}')$ is then obtained from $\mathrm{Winning}(\mathcal{A}')$ by projection on the component Σ. The automaton $\mathrm{Winning}(\mathcal{A}')$ itself is not concretely specified; it can be any deterministic automaton with the required property. We will define it in such a way that the eventual automaton Complement($\mathcal{A}'$) will coincide with the original automaton $\mathcal{A}$. It is convenient here to use a variant of a deterministic automaton where transitions leading to non-productive states are undefined, so that there is always *at most one* transition rather than exactly one. (So we momentarily suspend our proviso that all automata are trimmed, but the projection automaton will be trimmed again.) Let $\mathcal{A}_w = \langle \Sigma \times \{\mathtt{L},\mathtt{R}\}^{\Delta'}, Q, q_\mathrm{I}, \Delta_w, \Omega\rangle$; that is, the automaton shares the states and the priority function with $\mathcal{A}$. The transitions of $\mathcal{A}_w$ are defined as follows.

- For each $a \in \Sigma$ and $f \in \{\mathtt{L},\mathtt{R}\}^{\Delta'}$, we have $(\top, (a,f), \top, \top) \in \Delta_w$.
- If (q, a) are in disjunctive mode in $\mathcal{A}$, and $(q, a, q_\mathtt{L}, \top), (q, a, \top, q_\mathtt{R}) \in \Delta$, which implies that $(q, a, q_\mathtt{L}, q_\mathtt{R}) \in \Delta'$, and if $f\colon (q, a, q_\mathtt{L}, q_\mathtt{R}) \mapsto d$ (for $f \in \{\mathtt{L},\mathtt{R}\}^{\Delta'}$) then the unique transition of $\mathcal{A}_w$ from q over (a, f) is $(q, (a, f), q_\mathtt{L}, \top)$ or $(q, (a, f), \top, q_\mathtt{R})$ depending on whether $d = \mathtt{L}$ or $d = \mathtt{R}$.
- If (q, a) are in conjunctive mode in $\mathcal{A}$, and $(q, a, q_\mathtt{L}, q_\mathtt{R}) \in \Delta$, which implies that $(q, a, q_\mathtt{L}, \top)$, $(q, a, \top, q_\mathtt{R}) \in \Delta'$, and if moreover the selector f "behaves properly" in the sense that $f\colon (q, a, q_\mathtt{L}, \top) \mapsto \mathtt{L}$ and $f\colon (q, a, \top, q_\mathtt{R}) \mapsto \mathtt{R}$ then the unique transition of $\mathcal{A}_w$ from q over (a, f) is $(q, (a, f), q_\mathtt{L}, q_\mathtt{R})$. In all other cases, a transition from q over (a, f) is undefined.

Now, it is clear that an automaton Complement($\mathcal{A}_w$) obtained from $\mathcal{A}$ by replacing each transition $\big(q, (a, f), q_\mathtt{L}, q_\mathtt{R}\big)$ by $\big(q, a, q_\mathtt{L}, q_\mathtt{R}\big)$ (cf. Section 2) coincides with the original automaton $\mathcal{A}$. So the proof of Proposition 9 boils down to the following.

▷ Claim 11. The automaton $\mathcal{A}_w$ accepts a product tree $t \otimes \tau \in \mathrm{Tr}_{\Sigma\times\{\mathtt{L},\mathtt{R}\}^{\Delta'}}$ if and only if none of its paths is losing.

Proof. Let the automaton $\mathcal{A}_w$ accept a product tree $t \otimes \tau \in \mathrm{Tr}_{\Sigma\times\{\mathtt{L},\mathtt{R}\}^{\Delta'}}$ by a unique run τ. For the sake of contradiction, suppose that there exists a losing path $\big((a_n, f_n), d_n\big)_{n\in\omega}$ along with a sequence of transitions $\big(\delta'_n = (p_n, a_n, p_{\mathtt{L},n}, p_{\mathtt{R},n})\big)_{n\in\omega}$ of $\mathcal{A}'$, such that $p_0 = q_\mathrm{I}^{\mathcal{A}'}$, and $\forall n \in \omega,\ p_{n+1}{=}p_{d_n,n} \wedge d_n{=}f_n(\delta'_n)$. Let $\big(q_n\big)_{n\in\omega}$ be the sequence of states visited by the run τ on this path, i.e. $q_n = \tau(d_0 \dots d_{n-1})$. We verify by induction on n that $q_n = p_n \neq \top$. Clearly $q_n = q_\mathrm{I}^{\mathcal{A}} = q_\mathrm{I}^{\mathcal{A}'} = p_0 \neq \top$. Suppose the claim holds for n. If (q_n, a_n) is in conjunctive mode in $\mathcal{A}$ with the unique transition $(q_n, a_n, q_\mathtt{L}, q_\mathtt{R})$ (with $q_\mathtt{L}, q_\mathtt{R} \neq \top$) then (p_n, a_n) (with $p_n = q_n$) is in disjunctive mode in $\mathcal{A}'$, with two possible transitions. Let $\delta'_n = (p_n, a_n, p_{\mathtt{L},n}, p_{\mathtt{R},n}) = (q_n, a_n, q_{\mathtt{L},n}, \top)$; the other case is symmetric. It follows from the definition of $\mathcal{A}_w$ that in this case $f_n\colon \delta'_n \mapsto \mathtt{L}$, as otherwise the next transition would not be defined and the run τ would not be accepting. Clearly, the transition of $\mathcal{A}_w$ used at this point is $(q_n, (a_n, f_n), q_\mathtt{L}, q_\mathtt{R})$. Therefore, we have $d_{n+1} = \mathtt{L}$ and $q_{n+1} = q_\mathtt{L} = p_{n+1} \neq \top$, as required. If (q_n, a_n) is in disjunctive mode in $\mathcal{A}$ and hence $\delta'_n = (p_n, a_n, p_{\mathtt{L},n}, p_{\mathtt{R},n})$ is the unique transition of $\mathcal{A}'$ from $p_n = q_n$ over a_n, and moreover $f_n\colon \delta'_n \mapsto d_n$ then the transition of $\mathcal{A}_w$ from q_n over (a_n, f_n) is $(p_n, (a_n, f_n), p_{\mathtt{L},n}, \top)$ or $(q, (a_n, f_n), \top, p_{\mathtt{R},n})$ depending on whether $d_n = \mathtt{L}$ or $d = \mathtt{R}$, but in any case we have $q_{n+1} = q_\mathtt{L} = p_{n+1} \neq \top$, as required.

As the sequence of states $\big(q_n\big)_{n\in\omega}$ is parity accepting in $\mathcal{A}_w$ (hence also in $\mathcal{A}$), it cannot be at the same time parity accepting in $\mathcal{A}'$, which yields a contradiction.

Now let $t \otimes \tau \in \mathrm{Tr}_{\Sigma\times\{\mathtt{L},\mathtt{R}\}^{\Delta'}}$ be a product tree not accepted by $\mathcal{A}_w$. There are two possibilities: either the unique run of $\mathcal{A}_w$ on $t \otimes \tau$ is non-accepting, or there is no (complete) run at all, because the transitions are blocked at some place. We will show that in both cases there is a losing path in $t \otimes \tau$. Suppose first that in a run ρ of $\mathcal{A}_w$ on $t \otimes \tau$ there is a path $\big(q_n, d_n\big)_{n\in\omega}$, with $q_0 = q_\mathrm{I}^{\mathcal{A}}$ and $q_{n+1} = \rho(d_0 \dots d_n)$, such that the sequence $\big(q_n\big)_{n\in\omega}$ is parity rejecting. Thus in particular $q_n \neq \top$, for all n. Let $a_n = t(d_0 \dots d_{n-1})$ and $f_n = \tau(d_0 \dots d_{n-1})$. We will define inductively a sequence of transitions $\big(\delta'_n = (p_n, a_n, p_{\mathtt{L},n}, p_{\mathtt{R},n})\big)_{n\in\omega}$ of $\mathcal{A}'$

with $p_n = q_n$, which along with the sequence $(d_n)_{n\in\omega}$ will witness a losing path. As usual, we consider two cases. If (q_n, a_n) are in disjunctive mode in $\mathcal{A}$ then we let $\delta'_n \stackrel{\text{def}}{=} (q_n, a_n, q_{\mathtt{L}}, q_{\mathtt{R}})$, where the last is the unique transition of $\mathcal{A}'$ from q_n over a_n. Suppose further that $f_n\colon (q_n, a_n, q_{\mathtt{L}}, q_{\mathtt{R}}) \mapsto \mathtt{L}$ (the other case is symmetric). Then the transition used by the automaton $\mathcal{A}_w$ is $\big(q_n, (a_n, f_n), q_{\mathtt{L}}, \top\big)$, and we know that $q_{n+1} = q_{\mathtt{L}}$ (because it is different from $\top$), hence $d_n = \mathtt{L}$ and the local condition of a losing path is satisfied. If (q_n, a_n) are in conjunctive mode in $\mathcal{A}$ with the transition $(q_n, a_n, q_{\mathtt{L}}, q_{\mathtt{R}})$ then, since the transition of $\mathcal{A}_w$ over (a_n, f_n) is defined, we know that f_n "behaves properly", i.e. $f_n\colon (q_n, a_n, q_{\mathtt{L}}, \top) \mapsto \mathtt{L}$ and $f\colon (q_n, a_n, \top, q_{\mathtt{R}}) \mapsto \mathtt{R}$. Suppose $d_n = \mathtt{L}$ (the other case is symmetric), hence $q_{n+1} = q_{\mathtt{L}}$. We let $\delta'_n \stackrel{\text{def}}{=} (q_n, a_n, q_{\mathtt{L}}, \top)$; then again the local condition of a losing path is satisfied. Thus we have obtained a losing path in $t \otimes \tau$, as expected.

Now suppose that there is no run of $\mathcal{A}_w$ on $t \otimes \tau$. We may however define a *partial run*, i.e. a mapping $\rho\colon dom \to Q$, where $dom \subseteq \{\mathtt{L}, \mathtt{R}\}^*$ is closed under initial segment, such that $\rho(\epsilon) = q_{\mathrm{I}}^{\mathcal{A}}$, and whenever $\tau(v) = q$ and the transition $\big(q, (t(v), \rho(v)), q_{\mathtt{L}}, q_{\mathtt{R}}\big)$ of $\mathcal{A}_w$ is defined then $\rho(v\mathtt{L}) = q_{\mathtt{L}}$ and $\rho(v\mathtt{R}) = q_{\mathtt{R}}$. Since ρ is not complete, there must be a finite path $d_0, \ldots, d_{m-1}$, such that $\rho(d_0, \ldots, d_{m-1}) = q$, $t(d_0, \ldots, d_{m-1}) = a$, $\tau(d_0, \ldots, d_{m-1}) = f$, and there is no transition of $\mathcal{A}_w$ from q over (a, f). (It is possible that $m = 0$.) This means that (q, a) are in conjunctive mode in $\mathcal{A}$ with the transition $(q, a, q_{\mathtt{L}}, w_{\mathtt{R}})$, but f does not "behave properly"; for example $f\colon (q, a, q_{\mathtt{L}}, \top) \mapsto \mathtt{R}$ (the other case is symmetric). Let $q_i = \rho(d_0 \ldots d_{i-1})$, for $i = 0, \ldots, m$. We define a finite sequence $\delta'_0, \ldots, \delta'_{m-1}$ of transitions of $\mathcal{A}'$ corresponding to the path $d_0, \ldots, d_{m-1}$, exactly as in the previous case, satisfying the local condition of a losing path, with $p_i = q_i$, and hence $p_m = q_m = q$. Now, under the assumption that $f\colon (q, a, q_{\mathtt{L}}, \top) \mapsto \mathtt{R}$, we let $\delta'_m \stackrel{\text{def}}{=} (q, a, q_{\mathtt{L}}, \top)$, $d_m = \mathtt{R}$, and $p_{m+1} = \top$. Then we can clearly prolong this path to a losing path (with δ'_n, for $n \geq m+1$ being a transition for $\top$).

This concludes the proof of Claim 11, and hence also of Proposition 9. $\triangleleft$

5 Decidability of guidable index

In this section we provide the main result of the present article:

▶ **Theorem 12.** *Given a regular language of infinite trees $L \subseteq \mathrm{Tr}_\Sigma$ and an index $C = \{i, \ldots, j\}$ it is decidable whether there exists a guidable C-parity automaton which recognises L.*

Let $\mathcal{A} = \langle \Sigma, Q^{\mathcal{A}}, q_{\mathrm{I}}^{\mathcal{A}}, \Delta^{\mathcal{A}}, \Omega^{\mathcal{A}} \rangle$ be a guidable parity automaton for the given language L, which can be constructed as in Theorem 1.

Consider the following game, played between two players called $\exists$ and $\forall$. The positions of the game are pairs of states $Q^{\mathcal{A}} \times Q^{\mathcal{A}}$ of the automaton $\mathcal{A}$, the initial position (p_0, p'_0) is $(q_{\mathrm{I}}^{\mathcal{A}}, q_{\mathrm{I}}^{\mathcal{A}})$. At an nth round for $n = 0, 1, \ldots$ which starts in a position $(p_n, p'_n) \in Q^{\mathcal{A}} \times Q^{\mathcal{A}}$:

1. $\exists$ chooses a priority $c_n \in C$,
2. $\forall$ chooses a transition $\delta_n = (p_n, a_n, p_{\mathtt{L},n}, p_{\mathtt{R},n}) \in \Delta^{\mathcal{A}}$ from p_n over some letter $a_n \in \Sigma$,
3. $\exists$ chooses a transition $\delta'_n = (p'_n, a_n, p'_{\mathtt{L},n}, p'_{\mathtt{R},n}) \in \Delta^{\mathcal{A}}$ from p'_n over the same letter $a_n \in \Sigma$,
4. $\forall$ chooses a direction $d_n \in \{\mathtt{L}, \mathtt{R}\}$.

The next position of the game (p_{n+1}, p'_{n+1}) is $(p_{d_n,n}, p'_{d_n,n})$.

The winning condition for $\exists$ in that game is the conjunction of the following two conditions:

W1 If the sequence of states $(p_n)_{n\in\omega}$ is parity accepting in $\mathcal{A}$ then the sequence of priorities $(c_n)_{n\in\omega}$ must be also parity accepting.

W2 If the sequence of priorities $(c_n)_{n\in\omega}$ is parity accepting then the sequence of states $(p'_n)_{n\in\omega}$ is parity accepting in $\mathcal{A}$.

Clearly the winning condition of the game is ω-regular so the winner effectively has a finite memory winning strategy [3]. The rest of this section is devoted to a proof of the following proposition.

▶ **Proposition 13.** *The player $\exists$ wins the game above if and only if there exists a guidable C-parity automaton $\mathcal{B}$ which recognises the language $L = \mathrm{L}(\mathcal{A})$.*

▶ **Corollary 14.** *The problem of existence of a guidable C-parity automaton recognising a given regular language is decidable. Moreover, it is possible to effectively construct such an automaton whenever it exists.*

The construction of a guidable automaton for the language recognised by a given non-deterministic automaton may involve a double-exponential growth in the number of states [14, Proposition 4.8]. Therefore, already the size of the above game is in general double-exponential in the given representation of L. Since the index of the constructed automaton $\mathcal{A}$ is only single-exponential, there exists a double-exponential deterministic parity automaton with single-exponentially many priorities verifying the conjunction of the winning conditions W1 and W2 (cf. [6, Section 7]). It all implies that the complexity of the proposed algorithm is 2-EXPTIME.

Soundness

Assume that $\exists$ has a (finite memory) winning strategy in the game above. Such a strategy consists of a finite set of memory values M; the initial memory value $m_{\mathrm{I}} \in M$; a memory update function of the type:

$$\tau\colon Q^{\mathcal{A}} \times Q^{\mathcal{A}} \times M \times \Delta^{\mathcal{A}} \times \{\mathtt{L}, \mathtt{R}\} \to M,$$

which updates the memory value depending on the choices of the opponent in a given round; and two skolemised decision functions:

- $\overline{c}\colon Q^{\mathcal{A}} \times Q^{\mathcal{A}} \times M \to C$ which gives the choices of the priorities c_n depending only on the position of the game and the memory value in M,
- $\overline{\delta'}\colon Q^{\mathcal{A}} \times Q^{\mathcal{A}} \times M \times \Delta^{\mathcal{A}} \to \Delta^{\mathcal{A}}$ which gives the choices of the transitions δ'_n depending on the position of the game, the memory value in M, and the choice of the transition δ_n made by $\forall$ in the given round.

We construct a non-deterministic automaton $\mathcal{B}$ which guesses the choices of the transitions $\delta_n \in \Delta^{\mathcal{A}}$ and simulates the above strategy. Together with a definition of $\mathcal{B}$, we define a guiding function $g^{\mathcal{B}}\colon Q^{\mathcal{B}} \times \Delta^{\mathcal{A}} \to \Delta^{\mathcal{B}}$ that will be used to witness that $\mathcal{A} \hookrightarrow \mathcal{B}$.

Let the set of states of the automaton $\mathcal{B}$ be $Q^{\mathcal{A}} \times Q^{\mathcal{A}} \times M$. The initial state is $(q_{\mathrm{I}}^{\mathcal{A}}, q_{\mathrm{I}}^{\mathcal{A}}, m_{\mathrm{I}})$. The priority of a state (p, p', m) is $\overline{c}(p, p', m)$. Given a state (p, p', m) of $\mathcal{B}$ and a transition $\delta = (p, a, p_{\mathtt{L}}, p_{\mathtt{R}}) \in \Delta^{\mathcal{A}}$ we define $g^{\mathcal{B}}\big((p, p', m), \delta\big)$ as the transition $\Big((p, p', m), a, (p_{\mathtt{L}}, p'_{\mathtt{L}}, m_{\mathtt{L}}), (p_{\mathtt{R}}, p'_{\mathtt{R}}, m_{\mathtt{R}})\Big)$ such that $\overline{\delta'}(p, p', m, \delta) = \delta' = (p', a, p'_{\mathtt{L}}, p'_{\mathtt{R}}) \in \Delta^{\mathcal{A}}$ and for each direction $d \in \{\mathtt{L}, \mathtt{R}\}$ we have $\tau(p, p', m, \delta, d) = m_d$. The set of transitions of $\mathcal{B}$ consists of all the transitions $g^{\mathcal{B}}\big((p, p', m), \delta\big)$ for $(p, p', m) \in Q^{\mathcal{B}}$ and $\delta \in \Delta^{\mathcal{A}}$. This concludes the definition of the automaton $\mathcal{B}$.

In the following two proofs we will rely on the path-wise definition of when a guiding function preserves acceptance, see Remark 6.

The following fact follows directly from the assumption that we began with a winning strategy of $\exists$ and therefore its choices satisfy the winning condition W1.

▶ **Fact 15.** *The guiding function $g^{\mathcal{B}}$ defined above preserves acceptance. Thus, it witnesses that $\mathcal{A} \hookrightarrow \mathcal{B}$ and in particular $\mathrm{L}(\mathcal{A}) \subseteq \mathrm{L}(\mathcal{B})$.*

▶ **Lemma 16.** *We have $\mathcal{B} \hookrightarrow \mathcal{A}$ and in particular $\mathrm{L}(\mathcal{B}) \subseteq \mathrm{L}(\mathcal{A})$.*

Proof. We can use the decision function $\overline{\delta'}$ as a guide function $g^{\mathcal{A}} : Q^{\mathcal{A}} \times \Delta^{\mathcal{B}} \to \Delta^{\mathcal{A}}$. More formally, consider a transition γ of $\mathcal{B}$ that is of the form $g^{\mathcal{B}}\big((p, p', m), \delta\big)$ for some[1] $\delta \in \Delta^{\mathcal{A}}$. Let $g^{\mathcal{A}}(p', \gamma) = \overline{\delta'}(p, p', m, \delta)$; the remaining values of $g^{\mathcal{A}}(p'', \gamma)$ with the first argument p'' different than p' that comes from γ are irrelevant.

The fact that $g^{\mathcal{A}}$ preserves acceptance follows directly from the assumption that we began with a winning strategy of $\exists$ and therefore its choices satisfy the winning condition W2. ◀

The two above observations allow us to use Corollary 8 to learn the following fact.

▶ **Fact 17.** *The automaton $\mathcal{B}$ is guidable and recognises the language $L = \mathrm{L}(\mathcal{A})$.*

Since $\mathcal{B}$ is a C-parity automaton, the above fact concludes this direction of the proof.

Completeness

We will now show that if there exists a guidable C-parity automaton $\mathcal{B}$ which recognises L then $\exists$ has a winning strategy in the game above. Let $\mathcal{B} = \langle \Sigma, Q^{\mathcal{B}}, q_{\mathrm{I}}^{\mathcal{B}}, \Delta^{\mathcal{B}}, \Omega^{\mathcal{B}} \rangle$ be such a guidable automaton.

Fix two guiding functions $g^{\mathcal{A}} : Q^{\mathcal{A}} \times \Delta^{\mathcal{B}} \to \Delta^{\mathcal{A}}$ and $g^{\mathcal{B}} : Q^{\mathcal{B}} \times \Delta^{\mathcal{A}} \to \Delta^{\mathcal{B}}$, witnessing that $\mathcal{B} \hookrightarrow \mathcal{A}$ and $\mathcal{A} \hookrightarrow \mathcal{B}$, respectively.

Let the memory structure of the constructed strategy of $\exists$ consists of the set of states of $\mathcal{B}$. The initial memory value q_0 is $q_{\mathrm{I}}^{\mathcal{B}}$. In an nth round of the game that starts in a position $(p_n, p'_n) \in Q^{\mathcal{A}} \times Q^{\mathcal{A}}$ and with a memory value $q_n \in Q^{\mathcal{B}}$ let $\exists$ play as follows:

- $\exists$ plays the priority $c_n \stackrel{\text{def}}{=} \Omega^{\mathcal{B}}(q_n)$,
- $\forall$ plays a transition $\delta_n = (p_n, a_n, p_{\mathtt{L},n}, p_{\mathtt{R},n}) \in \Delta^{\mathcal{A}}$, which is mapped by $g^{\mathcal{B}}$ to a transition $\gamma = (q_n, a_n, q_{\mathtt{L},n}, q_{\mathtt{R},n}) \stackrel{\text{def}}{=} g^{\mathcal{B}}(q_n, \delta) \in \Delta^{\mathcal{B}}$,
- $\exists$ plays the transition $\delta'_n = (p'_n, a_n, p'_{\mathtt{L},n}, p'_{\mathtt{R},n}) \stackrel{\text{def}}{=} g^{\mathcal{A}}(p'_n, \gamma) \in \Delta^{\mathcal{A}}$,
- $\forall$ plays a direction $d_n \in \{\mathtt{L}, \mathtt{R}\}$,

and the next memory value q_{n+1} is $q_{d_n,n}$.

The following fact is an immediate consequence of the assumptions that the functions $g^{\mathcal{B}}$ and $g^{\mathcal{A}}$ map accepting runs into accepting runs, see Remark 6.

▶ **Lemma 18.** *The strategy defined above is winning for $\exists$.*

Proof. Consider an infinite play of the game where $\exists$ played according to the above defined strategy. First consider the winning condition W1. Assume that the sequence of states $(p_n)_{n\in\omega}$ is parity accepting in $\mathcal{A}$. Therefore, by the assumption on $g^{\mathcal{B}}$ we know that the sequence of states $(q_n)_{n\in\omega}$ is parity accepting in $\mathcal{B}$. But the choice of priorities c_n as $\Omega^{\mathcal{B}}(q_n)$ guarantees that the sequence of priorities $(c_n)_{n\omega}$ must also be parity accepting.

Now consider the winning condition W2 and assume that the sequence of priorities $(c_n)_{n\in\omega}$ is parity accepting. Therefore, the sequence of states $(q_n)_{n\in\omega}$ must be parity accepting in $\mathcal{B}$. Based on the assumption on $g^{\mathcal{A}}$ we know that the sequence of states $(p'_n)_{n\in\omega}$ must be parity accepting in $\mathcal{A}$. ◀

[1] In fact the transition δ is uniquely determined by γ.

6 Index transfer results

In this section we show results binding deterministic, game, and guidable indices of languages, as expressed by the following proposition.

▶ **Proposition 19.** *Assume that $L \subseteq \mathrm{Tr}_\Sigma$ can be recognised by some deterministic (resp. game) automaton and by some guidable C-parity automaton. Then L can be recognised by a deterministic (resp. game) C-parity automaton.*

First consider the case of deterministic automata. Let $\mathcal{D}$ be any deterministic automaton recognising L and let $\mathcal{A}$ be a guidable C-parity automaton recognising L. Let $g^{\mathcal{A}} \colon Q^{\mathcal{A}} \times \Delta^{\mathcal{D}} \to \Delta^{\mathcal{A}}$ be a guiding function witnessing that $\mathcal{D} \hookrightarrow \mathcal{A}$.

Consider the automaton $\mathcal{B}$ constructed as a product of $\mathcal{D}$ and $\mathcal{A}$. More formally, the set of states of $\mathcal{B}$ is $Q^{\mathcal{D}} \times Q^{\mathcal{A}}$; the initial state of $\mathcal{B}$ is $(q_{\mathrm{I}}^{\mathcal{D}}, q_{\mathrm{I}}^{\mathcal{A}})$; the priority function is defined as $\Omega^{\mathcal{B}}(q,p) = \Omega^{\mathcal{A}}(p)$; and the transitions of $\mathcal{B}$ are of the form

$$\big((q,p), a, (q_{\mathtt{L}}, p_{\mathtt{L}}), (q_{\mathtt{R}}, p_{\mathtt{R}})\big) \in \Delta^{\mathcal{B}}$$

where $\gamma = (q, a, q_{\mathtt{L}}, q_{\mathtt{R}})$ is the unique transition of $\mathcal{D}$ from q over a and $\delta = (p, a, p_{\mathtt{L}}, p_{\mathtt{R}}) \stackrel{\text{def}}{=} g^{\mathcal{A}}(p,\gamma) \in \Delta^{\mathcal{A}}$ is the transition given by the guiding function $g^{\mathcal{A}}$. Directly from the definition we see that $\mathcal{B}$ is a deterministic C-parity automaton. Therefore, the following fact concludes the proof of Proposition 19 in the case of deterministic automata.

▶ **Lemma 20.** $\mathrm{L}(\mathcal{B}) = L$*.*

Proof. The fact that $\mathrm{L}(\mathcal{B}) \subseteq \mathrm{L}(\mathcal{A})$ is immediate, as each accepting run of $\mathcal{B}$ encodes an accepting run of $\mathcal{A}$ over the same tree. On the other hand, if $t \in \mathrm{L}(\mathcal{D})$ then the assumptions on $g^{\mathcal{A}}$ imply that the unique run of $\mathcal{B}$ over t must be accepting. ◀

We now move to the proof of Proposition 19 in the case of game automata. Similarly as before, take any game automaton $\mathcal{D}$ recognising L and let $\mathcal{A}$ be a guidable C-parity automaton recognising L.

We first modify the automaton $\mathcal{A}$ by adding an additional state $\top$ of even priority and a transition of the form $(\top, a, \top, \top)$ for each letter $a \in \Sigma$. Now, for each state $p \in Q^{\mathcal{A}} - \{\top\}$ such that $\mathrm{L}(\mathcal{A}, p) = \mathrm{Tr}_\Sigma$, remove this state and replace each occurrence of p by $\top$ in all the transitions of $\mathcal{A}$. Clearly, these modifications do not change the language recognised by $\mathcal{A}$. Moreover, if the original automaton was guidable then the new one is also guidable. For the sake of simplicity we assume that from this moment on $\mathcal{A}$ denotes the modified automaton. Let $g^{\mathcal{A}} \colon Q^{\mathcal{A}} \times \Delta^{\mathcal{D}} \to \Delta^{\mathcal{A}}$ be a guiding function witnessing that $\mathcal{D} \hookrightarrow \mathcal{A}$.

The automaton $\mathcal{B}$ is defined analogously as in the deterministic case, as a product of the automata $\mathcal{D}$ and $\mathcal{A}$ using the guiding function $g^{\mathcal{A}}$. We additionally restrict the set of states of $\mathcal{B}$ to those that can be reached from the initial state $(q_{\mathrm{I}}^{\mathcal{D}}, q_{\mathrm{I}}^{\mathcal{A}})$ using the transitions of $\mathcal{B}$.

▷ Claim 21. If a state $(\top, p)$ is reachable from the initial state of $\mathcal{B}$ by the transitions of $\mathcal{B}$ then $\mathrm{L}(\mathcal{A}, p) = \mathrm{Tr}_\Sigma$, which means that $p = \top$.

Proof. Since all the transitions of $\mathcal{B}$ follow the guiding function $g^{\mathcal{A}}$, whenever a state (q,p) is reachable in the automaton $\mathcal{B}$, we know that it is a winning position of $\exists$ in the weak inclusion game $\mathcal{G}_{\mathrm{guide}}(\mathcal{D}, \mathcal{A})$, see Lemma 5. This guarantees that $\mathrm{L}(\mathcal{D}, q) \subseteq \mathrm{L}(\mathcal{A}, p)$, and as $\mathrm{L}(\mathcal{D}, \top) = \mathrm{Tr}_\Sigma$, we know that $\mathrm{L}(\mathcal{A}, p) = \mathrm{Tr}_\Sigma$. ◁

The above claim implies that $\mathcal{B}$ is in fact a game automaton with the shape of transitions inherited from $\mathcal{D}$ and $(\top, \top)$ playing the role of the state $\top$ of $\mathcal{B}$. Since the priority function $\Omega^{\mathcal{B}}$ is again inherited from $\mathcal{A}$, $\mathcal{B}$ is a C-parity automaton.

It remains to show that $\mathrm{L}(\mathcal{B}) = L$. Similarly as in the deterministic case, the inclusion $\mathrm{L}(\mathcal{B}) \subseteq \mathrm{L}(\mathcal{A})$ is immediate as each accepting run of $\mathcal{B}$ encodes an accepting run of $\mathcal{A}$. On the other hand, the inclusion $\mathrm{L}(\mathcal{D}) \subseteq \mathrm{L}(\mathcal{B})$ is direct from the assumptions on $g^{\mathcal{A}}$ – each accepting run of $\mathcal{D}$ is mapped by $g^{\mathcal{A}}$ into an accepting run of $\mathcal{B}$.

This concludes the proof of Proposition 19.

Consequences

By combining Proposition 19 and Proposition 9 (or by direct construction) we can observe the following.

▶ Remark 22. If $L \subseteq \mathrm{Tr}_\Sigma$ is recognised by some deterministic automaton and by some game C-parity automaton then L can be recognised by a deterministic C-parity automaton.

Thus the stratification from (1) preserves indices of the languages: if a language happens to be recognisable by a less expressive automaton then its parity index is the same from the point of view of both less and more expressive classes.

On the other hand, it is known that there is no such transfer when moving between deterministic and (general) non-deterministic automata. Indeed, we have the following property on ω-words.

▶ **Fact 23** ([22]). *For each $C = \{i, \ldots, j\} \subseteq \omega$ there exists a regular language of ω-words, which can be recognised by a non-deterministic $\{1, 2\}$-parity (Büchi) ω-word automaton, but not by any deterministic C-parity ω-word automaton.*

This property can be easily shifted to infinite trees (e.g., by considering the set of all trees whose leftmost branch belongs to a given regular language of ω-words). Thus we obtain the following consequence of Proposition 19.

▶ **Corollary 24.** *For each $C = \{i, \ldots, j\} \subseteq \omega$ there exists a regular tree language L which can be recognised by some deterministic automaton and some non-deterministic $\{1, 2\}$-parity automaton but not by any deterministic, game, nor guidable C-parity automaton.*

7 Conclusions

The present work is focused on the class of guidable automata. We show that they syntactically extend the previously studied classes of deterministic and game automata. Moreover, we provide an algorithm solving the guidable index problem. Finally, we show that for the three considered classes of deterministic, game, and guidable automata, the index can be transferred. As a negative consequence of this fact (Corollary 24) we show that there is no correspondence between the general non-deterministic index of a regular tree language and its guidable index.

Although these results do not bring a direct progress in the general non-deterministic index problem; we hope that they may be useful in at least two ways.

First, we believe that the class of guidable automata is worth separate attention. As our new results indicate, this class of tree automata is tractable and extends the previously considered classes of *structurally simple* automata: deterministic and game. This is especially important as, contrarily to the other two classes, guidable automata are expressively complete for all regular tree languages.

Second, the present results indicate a new way of approaching the index problems, by providing new game-based techniques. In particular, we believe that the interplay between the two sequences of transitions, δ_n and δ'_n, constructed in the game from Section 5, gives some insight on ways of forcing the players to witness the existence of certain objects.

References

1 Julian Bradfield. Simplifying the modal mu-calculus alternation hierarchy. In *STACS*, pages 39–49, 1998.

2 Julius Richard Büchi. On a decision method in restricted second-order arithmetic. In *Proc. 1960 Int. Congr. for Logic, Methodology and Philosophy of Science*, pages 1–11, 1962.

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

4 Cristian S. Calude, Sanjay Jain, Bakhadyr Khoussainov, Wei Li, and Frank Stephan. Deciding parity games in quasipolynomial time. In *49th Annual ACM STOC*, pages 252–263, 2017.

5 Lorenzo Clemente and Michał Skrzypczak. Deterministic and game separability for regular languages of infinite trees. accepted to ICALP, 2021.

6 Lorenzo Clemente and Michał Skrzypczak. Deterministic and game separability for regular languages of infinite trees, 2021. `arXiv:2105.01137`.

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

8 Jacques Duparc, Alessandro Facchini, and Filip Murlak. Linear game automata: Decidable hierarchy problems for stripped-down alternating tree automata. In *CSL*, pages 225–239, 2009.

9 E. Allen Emerson and Charanjit S. Jutla. The complexity of tree automata and logics of programs (extended abstract). In *29th FOCS*, pages 328–337, 1988.

10 E. Allen Emerson and Charanjit S. Jutla. Tree automata, mu-calculus and determinacy. In *FOCS*, pages 368–377, 1991.

11 Alessandro Facchini, Filip Murlak, and Michał Skrzypczak. Index problems for game automata. *ACM Transactions on Computational Logic*, 17(4):24:1–24:38, 2016.

12 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.

13 Yuri Gurevich and Leo Harrington. Trees, automata, and games. In *STOC*, pages 60–65, 1982.

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

15 Andrzej W. Mostowski. Regular expressions for infinite trees and a standard form of automata. In *Symposium on Computation Theory*, pages 157–168, 1984.

16 Damian Niwiński. On fixed-point clones. In *ICALP*, pages 464–473, 1986.

17 Damian Niwiński. Fixed point characterization of infinite behavior of finite-state systems. *Theoretical Computer Science*, 189(1–2):1–69, 1997.

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

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

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

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

22 Klaus Wagner. On ω-regular sets. *Information and Control*, 43(2):123–177, 1979.