On the weak index problem for game automata[⋆]

Alessandro Facchini[1], Filip Murlak[2], and Michał Skrzypczak[2]

[1] IDSIA, Switzerland
[2] University of Warsaw, Poland

Abstract. Game automata are known to recognise languages arbitrarily high in both the non-deterministic and alternating Rabin–Mostowski index hierarchies. Recently it was shown that for this class both hierarchies are decidable. Here we complete the picture by showing that the weak index hierarchy is decidable as well. We also provide a procedure computing for a game automaton an equivalent weak alternating automaton with the minimal index and a quadratic number of states. As a by-product we obtain that, as for deterministic automata, the weak index and the Borel rank coincide.

1 Introduction

Finite state automata running over trees constitute one of the main tools in the theory of verification and model-checking. In the latter, for instance, the model-checking problem is reduced to the non-emptiness problem for automata by translating the given formula into an automaton recognizing its models. The practical applicability of the automata-based approach thus relies on the possibility of being able to simplify the considered finite state machine.

In virtue of the trade off between expressibility and simplicity they present, weak alternating automata have emerged as a very appealing class of automata. Formally introduced in [MSS86], they are known to capture regular properties of infinite trees that are both Büchi and co-Büchi-recognizable, and to be expressively complete with respect to weak monadic second order logic [Rab70] and the alternation free fragment of the modal μ-calculus [AN92]. Because of their special structure, weak alternating automata have attractive computational properties. The corresponding non-emptiness problem can be for instance solved in linear time [BVW94], yielding an efficient (linear time) automata-based model checking algorithm for CTL. On the other hand, given a non-deterministic Büchi tree automaton A and another one recognizing its complement, [KV99] provide a translation of A into an equivalent weak alternating automaton with a quadratic number of states.

We can however look for more refined simplification procedures in which the parameter in the definition of an automaton reflecting the complexity of the recognised language is also taken into account. From this perspective, a

[⋆] The authors have been supported by Foundation for Polish Science grant Homing Plus no. 2012-5/1.

measure that has shown both practical and theoretical importance is the Rabin–Mostowski index, which measures the nesting of positive and negative conditions in the run of an automaton. The index orders tree automata into a hierarchy that was proved strict for deterministic [Wag79], non-deterministic [Niw86], alternating [Bra96], and weak alternating automata [Mos91b]. Computing the least possible index for a given regular language is called the index problem.

The only case for which this problem is know to have a solution for each of the four aforementioned modes is when the input language is deterministic [NW98,NW05,NW03,Mur08]. In [FMS13], it was shown that for the class of game automata (the closure of the class of deterministic automata under complementation and substitution) the non-deterministic and alternating index problems are solvable. The deterministic index problem being already solved by [NW98], the only case left is the weak index, known to coincide with the quantifier alternation depth for the weak monadic second order logic [Mos91b].

In this paper we show that the weak index problem is solvable for game automata by providing an effective translation to a weak alternating automaton with a quadratic number of states and the minimal index. As corollary of the result, we also obtain that, as for the class of deterministic automata [Mur08], for this class too the weak index and the Borel rank coincide.

2 Preliminaries

Trees. For a function f we write $\mathrm{dom}(f)$ for the domain of f and $\mathrm{rg}(f)$ for the range of f. For a finite alphabet A, we denote by PTr_A the set of partial trees over A, i.e., functions $t\colon \mathrm{dom}(t) \to A$ from a prefix-closed subset $\mathrm{dom}(t) \subseteq \{\mathtt{L},\mathtt{R}\}^*$ to A. By Tr_A we denote the set of *total* trees, i.e., trees t such that $\mathrm{dom}(t) = \{\mathtt{L},\mathtt{R}\}^*$. For a direction $d \in \{\mathtt{L},\mathtt{R}\}$ by $\bar{d}$ we denote the opposite direction. For $v \in \mathrm{dom}(t)$, $t{\restriction}_v$ denotes the subtree of t rooted at v. The sequences $u, v \in \{\mathtt{L},\mathtt{R}\}^*$ are naturally ordered by the prefix relation: $u \preceq v$ if u is a prefix of v.

A tree that is not total contains *holes.* A *hole* of tree t is a minimal sequence $h \in \{\mathtt{L},\mathtt{R}\}^*$ that does not belong to $\mathrm{dom}(t)$. By $\mathrm{holes}(t) \subseteq \{\mathtt{L},\mathtt{R}\}^*$ we denote the set of holes of tree t. If h is a hole of $t \in \mathrm{PTr}_A$, for $s \in \mathrm{PTr}_A$ we define the partial tree $t[h := s]$ obtained by putting the root of s into the hole h of t.

Games. A *parity game* G is a tuple $\langle V = V_\exists \cup V_\forall, v_I, E, \Omega\rangle$, where V is a countable *arena*; $V_\exists, V_\forall \subseteq V$ are positions of the game *belonging*, respectively, to player $\exists$ and player $\forall$, $V_\exists \cap V_\forall = \emptyset$; $v_I \in V$ is the initial position of the game; $E \subseteq V \times V$ is the transition relation; $\Omega\colon V \to \{i, \ldots, j\} \subseteq \mathbb{N}$ is a *priority function.* We assume that all parity games are finitely branching (for each $v \in V$ there are only finitely many $u \in V$ such that $(v, u) \in E$), and that there are no dead-ends (for each $v \in V$ there is at least one $u \in V$ such that $(v, u) \in E$).

A *play* in game G is an infinite sequence π of positions starting from v_I. Play π is *winning* for $\exists$ if $\liminf_{n\to\infty} \Omega(\pi(n))$ is even. Otherwise π is winning for $\forall$.

A (positional) *strategy* σ for a player $P \in \{\exists, \forall\}$ in a game G is defined as usual, as a function assigning to every P's position $v \in V_P$ the chosen successor

$\sigma(v) \in V$ such that $(v, \sigma(v)) \in E$. A play π *conforms to* σ if whenever π visits a vertex $v \in V_P$ then the next position of π is $\sigma(v)$. We say that a strategy σ is *winning* for P if every play conforming to σ is winning for P. In each parity game one of the players has a (positional) winning strategy [EJ91,Mos91a].

Automata. An alternating automaton $\mathcal{A}$ is a tuple $\langle A, Q, \delta, \Omega \rangle$, where A is a finite alphabet, Q is a finite set of states, $\Omega \colon Q \to \mathbb{N}$ is a function assigning to each state of $\mathcal{A}$ its priority, and δ assigns to each pair $(q, a) \in Q \times A$ the transition $b = \delta(q, a)$ built using the grammar

$$b \ ::= \ \top \ \big| \ \bot \ \big| \ (q, d) \ \big| \ b \vee b \ \big| \ b \wedge b \tag{1}$$

for states $q \in Q$ and directions $d \in \{\mathtt{L}, \mathtt{R}\}$.

For an alternating automaton $\mathcal{A}$, a state $q_I \in Q$, and a tree $t \in \mathrm{Tr}_A$ we define the game $\mathbf{G}(\mathcal{A}, t, q_I)$ as follows:

- $V = \mathrm{dom}(t) \times (B \uplus Q)$, where B is the set of all formulae generated by (1); positions of the form $(v, b_1 \vee b_2)$ belong to $\exists$ and the remaining ones to $\forall$; [3] the initial position is $v_I = (\epsilon, q_I)$;
- E contains the following pairs (for all $v \in \mathrm{dom}(t)$):
  $\big((v, b), (v, b)\big)$ for $b \in \{\top, \bot\}$,
  $\big((v, b), (v, b_i)\big)$ for $b = b_1 \wedge b_2$ or $b = b_1 \vee b_2$,
  $\big((v, (q, d)), (vd, q)\big)$ for $d \in \{\mathtt{L}, \mathtt{R}\}$, $q \in Q$,
  $\big((v, q), (v, \delta(q, t(v)))\big)$ for $q \in Q$;
- $\Omega(v, \top) = 0$, $\Omega(v, \bot) = 1$, $\Omega(v, q) = \Omega_{\mathcal{A}}(q)$ for $q \in Q$, $v \in \mathrm{dom}(t)$, and for other positions Ω is $\max(\mathrm{rg}(\Omega_{\mathcal{A}}))$, where $\Omega_{\mathcal{A}}$ is the priority function of $\mathcal{A}$.

An automaton $\mathcal{A}$ *accepts* a tree $t \in \mathrm{Tr}_A$ from $q_I \in Q$ if $\exists$ has a winning strategy in $\mathbf{G}(\mathcal{A}, t, q_I)$. By $\mathrm{L}(\mathcal{A}, q_I)$ we denote the set of trees accepted by automaton $\mathcal{A}$ from state q_I. Automaton $\mathcal{A}$ *recognises* a language $L \subseteq \mathrm{Tr}_A$ if $\mathrm{L}(\mathcal{A}, q_I) = L$ for some $q_I \in Q$.

The *(Rabin–Mostowski) index* of an automaton $\mathcal{A}$ is the pair (i, j) where i is the minimal and j is the maximal priority of the states of $\mathcal{A}$ ($\bot$ and $\top$ are counted as additional looping states with odd and even priority, respectively). In that case $\mathcal{A}$ is called an (i, j)-automaton.

An automaton $\mathcal{A}$ is *deterministic* if all its transitions are deterministic, i.e., of the form $\top$, $\bot$, (q_d, d), or $(q_{\mathtt{L}}, \mathtt{L}) \wedge (q_{\mathtt{R}}, \mathtt{R})$, for $d \in \{\mathtt{L}, \mathtt{R}\}$; $\mathcal{A}$ is *non-deterministic* if its transitions are (multifold) disjunctions of deterministic transitions.

An automaton $\mathcal{A}$ is *weak* if whenever $\delta(q, a)$ contains a state q' then $\Omega(q) \leq \Omega(q')$. For weak automata, allowing transitions $\top$ or $\bot$ interferes with the index much more then for strong automata: essentially, it adds one more change of priority. To reflect this, when defining the index of the automaton, we count $\bot$ and $\top$ as additional looping states with priorities assigned so that the weakness condition above is satisfied: $\bot$ gets the lowest *odd* priority ℓ such that $\bot$ is accessible only from states of priority at most ℓ, and dually for $\top$. That is, if the

[3] Positions $(v, (q, d)), (v, q), (v, \bot), (v, \top)$ offer no choice, so their owner is irrelevant.

automaton uses priorities $i, i+1, \ldots, 2k-1$, we can use $\bot$ for free (with priority $2k-1$), but for $\top$ we may need to pay with an additional priority $2k$, yielding index $(i, 2k)$. To emphasise the fact that an automaton in question is weak, we often call its index the *weak index.*

Game automata. In this work we study so-called *game automata*, i.e., alternating automata with transitions of the following forms:

$$\top\,, \quad \bot\,, \quad (q_d, d)\,, \quad (q_{\mathtt{L}}, \mathtt{L}) \vee (q_{\mathtt{R}}, \mathtt{R})\,, \quad (q_{\mathtt{L}}, \mathtt{L}) \wedge (q_{\mathtt{R}}, \mathtt{R})$$

for $d \in \{\mathtt{L}, \mathtt{R}\}$ and $q_{\mathtt{L}}, q_{\mathtt{R}} \in Q$.

The class of languages recognized by game automata is closed under complementation: the usual complementation procedure of increasing the priorities by one and swapping existential and universal transitions works. However it is neither closed under union nor intersection. For instance, let $L_\sigma = \{t \in T_{\{a,b\}} \colon t(\mathtt{L}) = t(\mathtt{R}) = \sigma\}$. Obviously, L_a and L_b are recognisable by game automata, but $L_a \cup L_b$ is not. Note that the last example also shows that game automata do not recognise all regular languages. On the other hand they extend across the whole alternating index hierarchy [FMS13].

The main similarity between game automata and deterministic automata is that their acceptance can be expressed in terms of *runs*, which are relabellings of input trees induced uniquely by transitions. For a game automaton $\mathcal{A}$ and an initial state q_I, with each partial tree t one can associate the *run*

$$\rho(\mathcal{A}, t, q_I) \colon \mathrm{dom}(t) \cup \mathrm{holes}(t) \to Q^{\mathcal{A}} \cup \{\top, \bot, *\}$$

such that $\rho(\varepsilon) = q_I$ and for all $v \in \mathrm{dom}(t)$, if $\rho(v) = q$, $\delta(q, t(v)) = b_v$, then

- if b_v is $(q_{\mathtt{L}}, \mathtt{L}) \vee (q_{\mathtt{R}}, \mathtt{R})$ or $(q_{\mathtt{L}}, \mathtt{L}) \wedge (q_{\mathtt{R}}, \mathtt{R})$, then $\rho(v\mathtt{L}) = q_{\mathtt{L}}$ and $\rho(v\mathtt{R}) = q_{\mathtt{R}}$;
- if $b_v = (q_d, d)$ for some $d \in \{\mathtt{L}, \mathtt{R}\}$, then $\rho(vd) = q_d$ and $\rho(v\bar{d}\,) = *$;
- if $b_v = \bot$ then $\rho(v\mathtt{L}) = \rho(v\mathtt{R}) = \bot$, and dually for $\top$;

and if $\rho(v) \in \{\top, \bot, *\}$, then $\rho(v\mathtt{L}) = \rho(v\mathtt{R}) = *$. Observe that $\rho(v)$ is uniquely determined by the labels of t on the path leading to v.

A run $\rho = \rho(\mathcal{A}, t, q_I)$ on a total tree t is naturally interpreted as a game $\mathbf{G}_\rho(\mathcal{A}, t, q_I)$ with positions $\mathrm{dom}(t) - \rho^{-1}(*)$, where edges follow the child relation and loop on $\rho^{-1}(\{\top, \bot\})$, priority of v is $\Omega^{\mathcal{A}}(\rho(v))$ with $\Omega^{\mathcal{A}}(\bot) = 1$, $\Omega^{\mathcal{A}}(\top) = 0$, and the owner of v is $\exists$ iff $\delta(\rho(v), t(v)) = (q_{\mathtt{L}}, \mathtt{L}) \vee (q_{\mathtt{R}}, \mathtt{R})$ for some $q_{\mathtt{L}}, q_{\mathtt{R}} \in Q^{\mathcal{A}}$. Clearly $\mathbf{G}_\rho(\mathcal{A}, t, q_I)$ is equivalent to $\mathbf{G}(\mathcal{A}, t, q_I)$. We say that ρ is *accepting*, if $\exists$ has a winning strategy in $\mathbf{G}_\rho(\mathcal{A}, t, q_I)$.

3 Computing the weak index

Informally, we would like to compute the weak index of a regular language given via a game automaton. There are two points to clarify here.

First, what is the weak index of a language L? We would like to say that it is the minimal index of a weak alternating automaton recognizing L, but how do we compare (i, j) and $(i+2, j+2)$? And what about (i, j) and $(i+1, j+1)$? We resolve this issue by looking at the classes of recognised languages. For $i \leq j \in \mathbb{N}$, let $\mathbf{RM}^w(i, j)$ be the class of languages recognised by weak alternating automata of index (i, j). We can always scale down the priorities so that the lowest one is either 0 or 1, so it suffices to consider the following classes of languages:

$$\mathbf{\Pi}_n^w = \mathbf{RM}^w(0, n)\,, \quad \mathbf{\Sigma}_n^w = \mathbf{RM}^w(1, n+1)\,, \quad \mathbf{\Delta}_n^w = \mathbf{\Pi}_n^w \cap \mathbf{\Sigma}_n^w$$

for $n \in \mathbb{N}$ ($n > 0$ in the case of $\mathbf{\Delta}_n^w$). These classes, naturally ordered by inclusion, constitute the *weak index hierarchy*. The *weak index of a language L* is the least class $\mathcal{C}$ in the weak index hierarchy such that $L \in \mathcal{C}$.

This brings us to the second issue: such class need not exist, because a game language need not be weakly recognisable. In [NW03] it is shown that a deterministic automaton recognises a weakly recognisable language if and only if it does not contain a forbidden pattern called *split*. The pattern is defined in terms of *paths* in the automaton, that is, sequences of states such that each state occurs in some transition for its predecessor in the sequence. A *split* consists of a branching transition $\delta(q, a) = (q_{\mathtt{L}}, \mathtt{L}) \wedge (q_{\mathtt{R}}, \mathtt{R})$, and paths from $q_{\mathtt{L}}$ to q and from $q_{\mathtt{R}}$ to q, one with odd minimal priority j_1, the other with even minimal priority j_2, satisfying $j_1 < j_2$. Since game automata are closed under dualisation, we must also be prepared for the *dual split*, which is defined like the split, except that the transition is controlled by $\exists$, i.e., $\delta(q, a) = (q_{\mathtt{L}}, \mathtt{L}) \vee (q_{\mathtt{R}}, \mathtt{R})$, and the minimal priorities satisfy $j_1 > j_2$. The following is an immediate corollary from the proof of the result of [NW03].

Fact 1. *If a game automaton $\mathcal{A}$ contains a split or a dual split reachable from state p, then the language* $\mathrm{L}(\mathcal{A}, p)$ *is not weakly recognisable.*

Now we can properly formulate our main result.

Theorem 1. *For a game automaton $\mathcal{A}$ with n states and a state q of $\mathcal{A}$, if $\mathcal{A}$ does not contain a split or a dual split reachable from q then* $\mathrm{L}(\mathcal{A}, q)$ *is weakly recognisable and its weak index can be computed effectively.*

Moreover, the witnessing automata with at most quadratic state-space can be constructed effectively within the time of solving the emptiness problem for $\mathcal{A}$.

The proof consists in a procedure computing the least class in the weak index hierarchy containing $\mathrm{L}(\mathcal{A}, q)$, denoted $\mathrm{wclass}(\mathcal{A}, q)$. The procedure works recursively on the DAG of strongly-connected components, or SCCs, of $\mathcal{A}$ (maximal sets of mutually reachable states). We identify SCCs of $\mathcal{A}$ with automata obtained by restricting $\mathcal{A}$ to the set of states in the SCC. Note that transitions originating in an SCC $\mathcal{B}$ can lead to states that are not in $\mathcal{B}$ any more. We call these states the *exits* of $\mathcal{B}$. Our procedure computes $\mathrm{wclass}(\mathcal{A}, q)$ based on $\mathrm{wclass}(\mathcal{A}, p)$ for exits p of the SCC $\mathcal{B}$ containing q. Those classes are aggregated in a way dependent on the internal structure of $\mathcal{B}$, or more precisely, on the way in which the state p is reachable from $\mathcal{B}$. The aggregation is be done by means of auxiliary operations on classes. Two most characteristic are

$$\left(\mathbf{\Pi}_{n-1}^w\right)^\exists = \left(\mathbf{\Delta}_n^w\right)^\exists = \left(\mathbf{\Sigma}_n^w\right)^\exists = \mathbf{\Sigma}_n^w\,, \quad \left(\mathbf{\Pi}_n^w\right)^\forall = \left(\mathbf{\Delta}_n^w\right)^\forall = \left(\mathbf{\Sigma}_{n-1}^w\right)^\forall = \mathbf{\Pi}_n^w\,.$$

We also use the bar notation for the dual classes,

$$\overline{\mathbf{\Pi}^w_n} = \mathbf{\Sigma}^w_n, \quad \overline{\mathbf{\Sigma}^w_n} = \mathbf{\Pi}^w_n, \quad \overline{\mathbf{\Delta}^w_n} = \mathbf{\Delta}^w_n,$$

and $\Phi \vee \Psi$ for the least class containing Φ and Ψ.

Before moving on to the details of the algorithm, we perform simple preprocessing. First, we eliminate trivial states. For each state q of $\mathcal{A}$ such that $\mathrm{L}(\mathcal{A}, q) = \emptyset$, we change transitions of the form $(q, d) \vee (q', d')$ to (q', d'), and transitions of the form $(q, d) \wedge (q', d')$ or (q, d) to $\bot$. Dually, if $\mathrm{L}(\mathcal{A}, q) = \mathrm{Tr}_A$, we replace $(q, d) \wedge (q', d')$ with (q', d'), and $(q, d) \vee (q', d')$ and (q, d) with $\top$. After this phase, the automaton contains only non-trivial states, that is, from each state q the automaton accepts some tree and rejects some other tree. The algorithmic cost of this preprocessing amounts to testing emptiness for $\mathrm{L}(\mathcal{A}, q)$ and $\mathrm{L}(\overline{\mathcal{A}}, q)$, where automaton $\overline{\mathcal{A}}$ is dualised automaton $\mathcal{A}$. Emptiness for game automata can be tested by determining the winner in a parity game similar to the game $\mathbf{G}(\mathcal{A}, t, q)$ discussed in Section 2. The difference is that the component t of the game is not present, and instead $\exists$ chooses the labels determining the transitions of $\mathcal{A}$. Since each tree node can be only reached in one way by the computation of $\mathcal{A}$, these choices are trivially consistent, and a tree together with an accepting run can be recovered from each winning strategy for $\exists$. The game uses the same priorities as $\mathcal{A}$, and its size is proportional to the size of $\mathcal{A}$.

The second stage is eliminating useless priorities. An n-component of automaton $\mathcal{A}$ is a maximal set of states that are mutually reachable via states of priority at least n. We say that an n-component is *non-trivial* if for some of its states p, q (not necessarily different), p occurs in some transition from q. Automaton $\mathcal{A}$ is *priority-reduced*, if for all $n > 0$, each n-component of $\mathcal{A}$ is non-trivial and contains a state of priority n. Each game automaton can be effectively transformed into an equivalent priority-reduced game automaton. To do it, we iteratively decrease priorities in the n-components of $\mathcal{A}$, for $n \geq 1$. We pick an n-component that is not priority-reduced; if it is trivial, we set all its priorities to $n-1$; if it is non-trivial but does not contain a state of priority n, we decrease all its priorities by 2 (this does not influence the recognised language). After finitely many steps the automaton is priority-reduced. Note that no trivial states are introduced.

Let us now describe the conditions that trigger applying the operations described above to previously computed classes. We begin with some shorthand notation. Let q', q'' be a pair of states in $\mathcal{B}$. Let $\max_\Omega(q' \to q'')$ be the maximal n such that there exists an n-path (a path with minimal priority n) from q' to q'' in $\mathcal{B}$. Observe that since $\mathcal{B}$ is an SCC, such n is well-defined (at least 0). Also, since the automaton is priority-reduced, for each $n' \leq \max_\Omega(q' \to q'')$ there is an n'-path from q' to q'' in $\mathcal{B}$.

An $\forall$-*branching* transition in $\mathcal{B}$ is a transition of the form $\delta(q', a) = (q_{\mathtt{L}}, \mathtt{L}) \wedge (q_{\mathtt{R}}, \mathtt{R})$ with all three states q', $q_{\mathtt{L}}$, $q_{\mathtt{R}}$ in $\mathcal{B}$; dually for $\exists$.

We say that a state p is $(\exists, n)$-*replicated by* $\mathcal{B}$ if there are states q', q'' in $\mathcal{B}$ and a letter a such that $\delta(q', a) = (q'', \mathtt{L}) \vee (p, \mathtt{R})$ (or symmetrically) and $\max_\Omega(q'' \to q') \geq n$. Dually, p is $(\forall, n)$-*replicated* if the transition above has the form $\delta(q', a) = (q'', \mathtt{L}) \wedge (p, \mathtt{R})$ (or the symmetrical).

We can now describe the procedure. By duality we can assume that the minimal priority in $\mathcal{B}$ is 0. If $\mathcal{A}$ contains no loop reachable from q, set $\mathrm{wclass}(\mathcal{A}, q) = \mathbf{\Delta}_1^w$. If it contains an accepting loop reachable from q, but no rejecting loop reachable from q, set $\mathrm{wclass}(\mathcal{A}, q) = \mathbf{\Pi}_1^w$. Symmetrically, if it contains a rejecting loop reachable from q, but no accepting loop reachable from q, set $\mathrm{wclass}(\mathcal{A}, q) = \mathbf{\Sigma}_1^w$. Otherwise, consider two cases.

Assume first that $\mathcal{B}$ contains no $\forall$-branching transition. In that case, for every transition $\delta(q, a)$ of $\mathcal{B}$ that is controlled by $\forall$, at most one of the successors of $\delta(q, a)$ is a state of $\mathcal{B}$. Hence, $\mathcal{B}$ can be seen as a co-deterministic tree automaton (exits are removed from the transitions; if both states in a transition are exits, the transition is set to $\bot$). Thus, the automaton $\bar{\mathcal{B}}$ dual to $\mathcal{B}$ is a deterministic tree automaton. For deterministic tree automata it is known how to compute the weak index [Mur08]. Set $\mathrm{wclass}(\mathcal{A}, q)$ to

$$\mathbf{\Delta}_2^w \vee \overline{\mathrm{wclass}(\bar{\mathcal{B}}, q)} \vee \bigvee_{p \in F} \mathrm{wclass}(\mathcal{A}, p) \vee \bigvee_{p \in F_{\exists,1}} \mathrm{wclass}(\mathcal{A}, p)^{\exists} \vee \bigvee_{p \in F_{\forall,0}} \mathrm{wclass}(\mathcal{A}, p)^{\forall} \quad (2)$$

where $F \subseteq Q^{\mathcal{A}}$ is the set of exits of $\mathcal{B}$, $F_{\exists,1} \subseteq F$ is the set of states $(\exists, 1)$-replicated by $\mathcal{B}$, and similarly for $F_{\forall,0}$.

Assume now that $\mathcal{B}$ does contain an $\forall$-branching transition. By the hypothesis of the theorem, for every $\forall$-branching transition $\delta(q', a) = (q_{\mathtt{L}}, \mathtt{L}) \wedge (q_{\mathtt{R}}, \mathtt{R})$ in $\mathcal{B}$, it must hold that $\max_{\Omega}(q_{\mathtt{L}} \to q') \leq 1$ and $\max_{\Omega}(q_{\mathtt{R}} \to q') = 0$, or symmetrically. We call a state q'' (either $q_{\mathtt{L}}$ or $q_{\mathtt{R}}$) in an $\forall$-branching transition *bad* if $\max_{\Omega}(q'' \to q') = 0$. Let $\mathcal{B}^-$ be obtained from $\mathcal{B}$ by replacing all these *bad* states in the $\forall$-branching transitions with $\top$, and let $\mathcal{A}^-$ be the automaton $\mathcal{A}$ with $\mathcal{B}$ replaced by $\mathcal{B}^-$ ($\mathcal{B}^-$ contains no $\forall$-branching transitions). Let us put

$$\mathrm{wclass}(\mathcal{A}, q) = \mathbf{\Delta}_2^w \vee \big(\mathrm{wclass}(\mathcal{A}^-, q)\big)^{\forall}. \quad (3)$$

4 On the correctness of the algorithm

To show correctness of the algorithm described in Section 3, we need to prove that $\mathrm{wclass}(\mathcal{A}, q) \leq \mathbf{RM}^w(i, j)$ if and only if $\mathrm{L}(\mathcal{A}, q)$ can be recognised by a weak alternating automaton of index (i, j). The left-to right part of this equivalence is proved by constructing an appropriate weak alternating automaton (see Appendix A); the construction is effective and involves only quadratic blow-up in the number of states, thus proving the additional claim of Theorem 1. We discuss in more detail the opposite implication, equivalently formulated as follows.

Lemma 1. *If* $\mathrm{wclass}(\mathcal{A}, q) \geq \mathbf{RM}^w(i, j)$ *then* $\mathrm{L}(\mathcal{A}, q)$ *cannot be recognised by a weak alternating automaton of index* $(i + 1, j + 1)$.

To prove it we use a topological argument, relying on the following simple observation [DM07], essentially proved already by Mostowski [Mos91b]. Let us assume the usual Cantor-like topology on the space of trees, with the open sets defined as arbitrary unions of sets of the form $\{t \in \mathrm{Tr}_A \mid t(v) = a\}$ for $v \in \{\mathtt{L}, \mathtt{R}\}^*$

and $a \in A$. Let $\mathbf{\Pi}^0_n$, $\mathbf{\Sigma}^0_n$, and $\mathbf{\Delta}^0_n$ be the finite Borel classes; that is, $\mathbf{\Sigma}^0_1$ is the class of the open sets, $\mathbf{\Pi}^0_n$ consists of the complements of sets from $\mathbf{\Sigma}^0_n$, $\mathbf{\Delta}^0_n = \mathbf{\Sigma}^0_n \cap \mathbf{\Pi}^0_n$, and $\mathbf{\Sigma}^0_{n+1}$ consists of countable unions of sets from $\mathbf{\Pi}^0_n$.

Fact 2. *If L is recognisable by a weak alternating automaton of index $(0, j)$ then $L \in \mathbf{\Pi}^0_j$. Dually, for index $(1, j+1)$, we have $L \in \mathbf{\Sigma}^0_j$.*

Thus, in order to show that a language is *not* recognisable by weak alternating automaton of index $(0, j)$ it is enough to show that it is not in $\mathbf{\Pi}^0_j$. This can be done by providing a continuous reduction of some language $K \notin \mathbf{\Pi}^0_j$ to L. By a *continuous reduction* of $K \subseteq \mathrm{Tr}_A$ to $L \subseteq \mathrm{Tr}_B$ we mean a continuous function $f\colon \mathrm{Tr}_A \to \mathrm{Tr}_B$ such that $f^{-1}(L) = K$. The fact that K can be continuously reduced to L is denoted by $K \leq_W L$, yielding so-called Wadge pre-order [Wad83]. The pre-order $\leq_W$ is consistent with the Borel hierarchy: for $K \leq_\mathrm{W} L$, if $L \in \mathcal{C}$ for some Borel class $\mathcal{C}$, then also $K \in \mathcal{C}$. By contraposition, if $K \notin \mathcal{C}$, then $L \notin \mathcal{C}$.

The yardstick languages we shall use, introduced by Skurczyński [Sku93], can be defined by means of two dual operations on tree languages.

Definition 1. *For $L \subseteq \mathrm{Tr}_A$ define*

$$L^\forall = \left\{ t \in \mathrm{Tr}_A \;\middle|\; \forall_{n\in\mathbb{N}}\, t{\restriction_{\mathtt{L}^n\mathtt{R}}} \in L \right\}, \quad L^\exists = \left\{ t \in \mathrm{Tr}_A \;\middle|\; \exists_{n\in\mathbb{N}}\, t{\restriction_{\mathtt{L}^n\mathtt{R}}} \in L \right\}.$$

It is straightforward to check that these operations are monotone with respect to the Wadge ordering; that is,

$$L \leq_\mathrm{W} M \text{ implies } L^\forall \leq_\mathrm{W} M^\forall \text{ and } L^\exists \leq_\mathrm{W} M^\exists.$$

Definition 2 ([Sku93])**.** *Consider the alphabet $\{\bot, \top\}$. Let*

$$S_{(0,1)} = \left\{ t \in \mathrm{Tr}_{\{\bot,\top\}} \;\middle|\; t(\epsilon) = \top \right\}^\forall, \quad S_{(1,2)} = \left\{ t \in \mathrm{Tr}_{\{\bot,\top\}} \;\middle|\; t(\epsilon) = \bot \right\}^\exists.$$

The remaining languages are defined inductively,

$$S_{(0,j+1)} = \left(S_{(1,j+1)}\right)^\forall, \quad S_{(1,j+1)} = \left(S_{(0,j-1)}\right)^\exists.$$

For notational convenience, let $S_{(0,0)} = \mathrm{Tr}_{\{\bot,\top\}}$ and $S_{(1,1)} = \emptyset$.

Note that the languages are dual to each other: $S_{(1,j+1)} = \mathrm{Tr}_{\{\bot,\top\}} - S_{(0,j)}$. A straightforward reduction shows that $S_{(i',j')} \leq_\mathrm{W} S_{(i,j)}$ whenever (i, j) is at least (i', j'). But the crucial property is the following.

Fact 3 ([Sku93])**.** *$S_{(0,n)} \in \mathbf{\Pi}^0_n - \mathbf{\Sigma}^0_n$ and $S_{(1,n+1)} \in \mathbf{\Sigma}^0_n - \mathbf{\Pi}^0_n$.*

Summing up, if $S_{(i,j)} \leq_W L$ then L is not recognisable by a weak alternating automaton of index $(i+1, j+1)$. Observe that the language $S_{(i,j)}$ can be recognised by a weak game automaton of index (i, j). One consequence of this is the strictness of the hierarchy.

Corollary 1. *The weak index hierarchy is strict, even when restricted to languages recognisable by game automata.*

Another consequence is that it is relatively easy to give the reductions we need to prove Lemma 1, summarised in the claim below (see Appendix B).

Lemma 2. *If $\mathrm{wclass}(\mathcal{A}, q) \geq \mathbf{RM}^w(i, j)$ then $S_{(i,j)} \leq_\mathrm{W} \mathrm{L}(\mathcal{A}, q)$.*

5 Conclusions

Game automata were originally introduced as the largest class extending deterministic automata (satisfying natural closure properties), such that substitution preserves the Wadge equivalence [DFM11]. Despite structural simplicity, they have enough expressive power to inhabit all levels of the non-deterministic index hierarchy and the alternating index hierarchy. In [FMS13] it was shown that these two hierarchies are decidable when the input language is recognized by a game automata.

So far, the only known class having all index problem decidable was the class of deterministic automata. In this paper we have shown that the same holds for game automata. This has been done by providing a procedure computing for a game automaton an equivalent weak alternating automaton with the minimal index and a quadratic number of states.

Another notable feature of tree languages recognised by deterministic automata is that within this class, the properties of being Borel and being weakly recognizable are coextensive. Since the former is decidable [NW03], the latter is also decidable. This correspondence can be made even more precise: for languages recognised by deterministic automata, the weak index and the Borel rank coincide [Mur08]. Notice that this implies that the Borel rank for deterministic languages is also decidable, a result originally proved in [Mur05]. As a corollary of the work presented in the previous sections, we obtain that the same is true for game automata.

Corollary 2. *Under restriction to languages recognised by game automata, the weak index hierarchy coincides with the Borel hierarchy, and both are decidable.*

Proof. From [Mur08], we know that if $\mathrm{wclass}(\mathcal{A}, q) \leq \mathbf{RM}^w(i, j)$ then $\mathrm{L}(\mathcal{A}, q) \leq_{\mathrm{W}} S_{(i,j)}$. The coincidence between weak index and Borel rank thence follows by applying Lemma 2 and Fact 3. Decidability is a consequence of Theorem 1. $\square$

This last result is yet another argument in support of the claim that all good properties enjoyed by languages recognised by deterministic automata are also enjoyed by languages recognised by game automata.

References

AN92. André Arnold and Damian Niwiński. Fixed point characterisation of weak monadic logic definable sets of trees. In *Tree Automata and Languages*, pages 159–188, 1992.

Bra96. Julian Bradfield. The modal mu-calculus alternation hierarchy is strict. In *CONCUR*, pages 233–246, 1996.

BVW94. Orna Bernholtz, Moshe Y. Vardi, and Pierre Wolper. An automata-theoretic approach to branching-time model checking (extended abstract). In *CAV*, pages 142–155, 1994.

DFM11. Jacques Duparc, Alessandro Facchini, and Filip Murlak. Definable operations on weakly recognizable sets of trees. In *FSTTCS*, pages 363–374, 2011.

DM07. Jacques Duparc and Filip Murlak. On the topological complexity of weakly recognizable tree languages. In *Fundamentals of Computation Theory, 16th International Symposium, FCT 2007, Budapest, Hungary, August 27-30, 2007, Proceedings*, pages 261–273, 2007.

EJ91. Allen Emerson and Charanjit Jutla. Tree automata, mu-calculus and determinacy. In *FOCS'91*, pages 368–377, 1991.

FMS13. Alessandro Facchini, Filip Murlak, and Michał Skrzypczak. Rabin–Mostowski index problem: A step beyond deterministic automata. In *LICS*, pages 499–508, 2013.

KV99. Orna Kupferman and Moshe Y. Vardi. The weakness of self-complementation. In *STACS*, pages 455–466, 1999.

Mos91a. Andrzej W. Mostowski. Games with forbidden positions. Technical report, University of Gdańsk, 1991.

Mos91b. Andrzej W. Mostowski. Hierarchies of weak automata and weak monadic formulas. *Theor. Comput. Sci.*, 83(2):323–335, 1991.

MSS86. David E. Muller, Ahmed Saoudi, and Paul E. Schupp. Alternating automata. the weak monadic theory of the tree, and its complexity. In Laurent Kott, editor, *ICALP*, volume 226 of *Lecture Notes in Computer Science*, pages 275–283, 1986.

Mur05. Filip Murlak. On deciding topological classes of deterministic tree languages. In *CSL 2005*, pages 428–441, 2005.

Mur08. Filip Murlak. Weak index versus Borel rank. In *STACS 2008*, volume 1 of *LIPIcs*, pages 573–584, 2008.

Niw86. Damian Niwiński. On fixed-point clones. In Laurent Kott, editor, *Automata, Languages and Programming*, volume 226 of *Lecture Notes in Computer Science*, pages 464–473. Springer Berlin Heidelberg, 1986.

NW98. Damian Niwiński and Igor Walukiewicz. Relating hierarchies of word and tree automata. In *STACS*, pages 320–331, 1998.

NW03. Damian Niwiński and Igor Walukiewicz. A gap property of deterministic tree languages. *Theor. Comput. Sci.*, 1(303):215–231, 2003.

NW05. Damian Niwiński and Igor Walukiewicz. Deciding nondeterministic hierarchy of deterministic tree automata. *Electr. Notes Theor. Comput. Sci.*, 123:195–208, 2005.

Rab70. Michael O. 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.

Sku93. Jerzy Skurczyński. The Borel hierarchy is infinite in the class of regular sets of trees. *Theoretical Computer Science*, 112(2):413–418, 1993.

Wad83. William Wadge. *Reducibility and determinateness in the Baire space.* PhD thesis, University of California, Berkeley, 1983.

Wag79. Klaus Wagner. On ω-regular sets. *Information and Control*, 43(2):123–177, 1979.

A Upper bounds

Lemma 3. *If* $\mathrm{wclass}(\mathcal{A}, q) \le \mathbf{RM}^w(i,j)$*, one can construct effectively a weak alternating automaton of index* (i,j) *with* $\mathcal{O}(|Q^{\mathcal{A}}|^2)$ *states, recognising* $\mathrm{L}(\mathcal{A}, q)$.

The algorithm never returns $\mathbf{\Pi}_0^w = \mathbf{RM}^w(0,0)$ nor $\mathbf{\Sigma}_0^w = \mathbf{RM}^w(1,1)$, so the lowest (i,j) to consider are $(0,1)$ and $(1,2)$. Assume $(i,j) = (0,1)$. Examining the algorithm we see that this happens only if no rejecting loop is reachable from state q. Since automaton $\mathcal{A}$ is priority-reduced, it means that $\mathcal{A}$ uses only priority 0. Hence, it is already a $(0,1)$ weak automaton (not $(0,0)$, because of possible $\perp$ transitions). For $(i,j) = (1,2)$ the argument is entirely analogous.

For higher indices we consider three cases, leading to three different constructions of weak alternating automata recognising $\mathrm{L}(\mathcal{A}, q)$.

$\mathcal{B}$ has no $\forall$-branching transitions, $(i,j) = (1,j)$, $j \ge 3$. In an initial part of the weak automaton recognising $\mathrm{L}(\mathcal{A}, q)$ the players declare whether during the play on a given tree they would leave component $\mathcal{B}$ or not. Since $\mathcal{B}$ has no $\forall$-branching transitions, as long as the play stays in $\mathcal{B}$, each choice of $\forall$ amounts to leaving $\mathcal{B}$ or staying in $\mathcal{B}$. Hence, each strategy of $\exists$ admits exactly one path staying in $\mathcal{B}$, finite or infinite. We first let $\exists$ declare $l_\exists \in \{\textit{leave}, \textit{stay}\}$: *leave* means that the path is finite, *stay* means that it is infinite.

- If $l_\exists = \textit{leave}$, we move to a copy of $\mathcal{B}$ with all the priorities set to 1. By Equation (2), for every exit f of $\mathcal{B}$ we have $\mathrm{wclass}(\mathcal{A}, f) \le \mathbf{RM}^w(1,j)$. Therefore, we can compose this copy of $\mathcal{B}$ with all the automata for $\mathrm{L}(\mathcal{A}, f)$ to obtain an automaton of index $(1,j)$, and we are done.
- Assume that $l_\exists = \textit{stay}$. Given the special shape of $\exists$'s strategies, this means that $\exists$ claims that the play will only leave $\mathcal{B}$ if at some point $\forall$ chooses an exit f in a transition whose other end is in $\mathcal{B}$. Since the minimal priority in $\mathcal{B}$ is 0, all these exists are $(\forall, 0)$-replicated. We check $\exists$'s claim by substituting all other exits in transitions with rejecting states, i.e. weak alternating automata of index $(3,3)$ (recall that j is at least 3). Thus, the only exits that are not substituted are the $(\forall, 0)$-replicated ones.

Now, assuming $l_\exists = \textit{stay}$, we ask $\forall$ whether he plans to take one of these exists: he declares $l_\forall \in \{\textit{leave}, \textit{stay}\}$, accordingly.

- $\underline{\text{If } l_\forall = \textit{stay}}$, the play moves to the weak alternating automaton of index $\mathrm{wclass}_{\mathrm{det}}(\bar{\mathcal{B}})$, corresponding to the co-deterministic automaton $\mathcal{B}$ with the remaining exits removed from transitions (they were only present in transitions of the form $(q_{\mathtt{L}}, \mathtt{L}) \wedge (q_{\mathtt{R}}, \mathtt{R})$, with the other state in $\mathcal{B}$).
- Assume that $l_\forall = \textit{leave}$. In that case we move to a copy of $\mathcal{B}$ with all the priorities set to 2. The only exits left are the $(\forall, 0)$-replicated ones. By Equation (2), for such exists f, $\mathrm{wclass}(\mathcal{A}, f) \le \mathbf{RM}^w(0, j-2)$: otherwise $\mathrm{wclass}(\mathcal{A}, f) \ge \mathbf{RM}^w(1, j-1)$, so $\big(\mathrm{wclass}(\mathcal{A}, p)\big)^\forall \ge \mathbf{RM}^w(0, j-1)$ and $\mathbf{RM}^w(0, j-1)$ is not smaller than $\mathbf{RM}^w(1, j)$. In particular, we can find a weak alternating automaton of index $(2,j)$ recognising $\mathrm{L}(\mathcal{A}, f)$. So the whole subautomaton is a weak alternating automaton of index $(2,j)$.

$\mathcal{B}$ has no $\forall$-branching transitions, $(i,j) = (0,j)$, $j \geq 2$. The simulation starts in a copy of $\mathcal{B}$ with all the priorities set to 0. If the play leaves $\mathcal{B}$ at this stage then we move to the appropriate automaton of index $(0,j)$. At any moment $\forall$ can pledge that:

- the play will no longer visit transitions $\delta(q',a)$ of the form $(f_{\mathtt{L}},\mathtt{L}) \wedge (f_{\mathtt{R}},\mathtt{R})$, $(f_{\mathtt{L}},\mathtt{L}) \vee (f_{\mathtt{R}},\mathtt{R})$, $(q_{\mathtt{L}},\mathtt{L}) \vee (f_{\mathtt{R}},\mathtt{R})$, $(f_{\mathtt{L}},\mathtt{L}) \vee (q_{\mathtt{R}},\mathtt{R})$, or $(q_{\mathtt{L}},\mathtt{L}) \vee (q_{\mathtt{R}},\mathtt{R})$, where $\max_{\Omega}(q_{\mathtt{L}} \to q') = \max_{\Omega}(q_{\mathtt{R}} \to q') = 0$ and $f_{\mathtt{L}}$, $f_{\mathtt{R}}$ are exits of $\mathcal{B}$;
- in the transitions he controls, he will always choose the state in $\mathcal{B}$, and win regardless of $\exists$'s choices.

If the play stays forever in $\mathcal{B}$ but $\forall$ is never able to make such a pledge, he loses by the parity condition — it means that infinitely many times a loop from $q_{\mathtt{L}} \to q'$ or $q_{\mathtt{R}} \to q'$ is taken with $\max_{\Omega}(q_d \to q') = 0$ therefore, the minimal priority occurring infinitely often is 0.

After $\forall$ has made the above pledge, $\exists$ has the following choices:

- She can challenge the first part of $\forall$'s pledge, declaring that at least one such transition is reachable. In that case we move to a copy of $\mathcal{B}$ with all the priorities set to 1 and all the transitions controlled by $\exists$. In this copy, reaching any of the disallowed transitions entails acceptance—the play immediately moves to a $(2,2)$ final component.
- She can accept the first part of $\forall$'s pledge.

After $\exists$ has accepted the first part of $\forall$'s pledge, we can assume that the rest of the game in $\mathcal{B}$ is a single infinite branch. Indeed, by the hypothesis of the theorem, for every $\exists$-branching transition $\delta(q',a) = (q_{\mathtt{L}},\mathtt{L}) \vee (q_{\mathtt{R}},\mathtt{R})$ in $\mathcal{B}$ it must hold that $\max_{\Omega}(q_{\mathtt{L}} \to q') = \max_{\Omega}(q_{\mathtt{R}} \to q') = 0$; otherwise, $\mathcal{B}$ would contain a dual split. Thus, no $\exists$-branching transition can be reached, and since $\mathcal{B}$ contains no $\forall$-branching transitions at all, the game can continue in $\mathcal{B}$ in only one way.

Now $\exists$ must challenge the second part of $\forall$'s pledge. We ask her whether she plans to leave $\mathcal{B}$ or not, and she declares $l_{\exists} \in \{\textit{leave}, \textit{stay}\}$.

- If $l_{\exists} = \textit{stay}$ then we proceed to the weak automaton of index $\mathrm{wclass}(\mathcal{B},q)$, corresponding to $\mathcal{B}$ treated as a co-deterministic automaton. We are only interested in the behaviour of this automaton over trees in which there is exactly one branch in $\mathcal{B}$, and it is infinite. Over such trees we want to make sure that neither players ever chooses to exit. This is already ensured: when $\mathcal{B}$ is turned into a co-deterministic tree automaton, the exits are simply removed from transitions (if both states are exits, the transition is changed to a transition to a $(2,2)$ automaton, but such transitions will never be used over trees we are interested in).
- If $l_{\exists} = \textit{leave}$ then we move to a copy of $\mathcal{B}$ with all the priorities set to 1. The only available exits of $\mathcal{B}$ in this copy are those in transitions of the form $\delta(q',q) = (q_{\mathtt{L}},\mathtt{L}) \vee (f,\mathtt{R})$ (or symmetrical) with $\max_{\Omega}(q_{\mathtt{L}} \to q') > 0$ (in other transitions the exits are removed, if both states are exits, they are replaced by a final $(2,2)$-component); therefore $\mathrm{wclass}(\mathcal{A},f) \leq \mathbf{RM}^{w}(1,j)$ and we can simulate it with a $(1,j)$-automaton.

$\mathcal{B}$ contains $\forall$-branching transitions If $\mathcal{B}$ contains an $\forall$-branching transition, the algorithm returns $\mathrm{wclass}(\mathcal{A}, q)$ of the form $\mathbf{RM}^w(0, j)$. Let us construct a weak automaton of index $(0, j)$ that recognises $\mathrm{L}(\mathcal{A}, q)$. The automaton starts in a copy of $\mathcal{B}$ with all the priorities set to 0. At any moment $\forall$ can declare that no-one will ever take any *bad* transition in $\mathcal{B}$. If he cannot make such a declaration, it means that $\exists$ can force infinitely many bad transitions to be taken, and she wins. After $\forall$ has made such declaration, we need to recognise the language $\mathrm{L}(\mathcal{A}^-, q)$ (note that the bad transitions in $\mathcal{A}^-$ are made directly losing for $\forall$). For this we can use a weak automaton of index $\mathrm{wclass}(\mathcal{A}^-) \le \mathbf{RM}^w(0, j)$, already constructed.

Constructed automaton has quadratic number of states. The preprocessing we make to guarantee that the automaton is priority-reduced does not increase the number of states. The resulting automaton consists of:

- a fixed number of copies of $\mathcal{B}$,
- a weak alternating automaton of index $\overline{\mathrm{wclass}_{\mathrm{det}}(\bar{\mathcal{B}})}$,
- a fixed number of states where players make decisions (e.g. $l_\forall \in \{\mathit{leave}, \mathit{stay}\}$),
- inductively constructed automata recognizing $\mathrm{L}(\mathcal{A}, f)$ for all exists f of $\mathcal{B}$.

By [Mur08, Theorem 5.5], the automaton in the second item has $\mathcal{O}(|Q^{\mathcal{B}}|^2)$ states. Hence, we inductively ensure the constructed automaton has $\mathcal{O}(|Q^{\mathcal{A}}|^2)$ states.

When $\mathcal{A}$ is priority-reduced with all the states productive, the rest of the construction is polynomial in the number of states of $\mathcal{A}$. Therefore, the whole construction can be done in the time of solving the emptiness and completeness problems of $\mathrm{L}(\mathcal{A}, q)$ for each state q of $\mathcal{A}$ separately.

B Lower bounds

Lemma 2. *If* $\mathrm{wclass}(\mathcal{A}, q) \ge \mathbf{RM}^w(i, j)$ *then* $S_{(i,j)} \le_{\mathrm{W}} \mathrm{L}(\mathcal{A}, q)$.

We prove this claim by induction on the structure of the DAG of SCCs of $\mathcal{A}$ reachable from q, following the cases of the algorithm just like for the upper bound. One of the cases is covered by the procedure for deterministic automata, which we use as a black box. But in order to prove Lemma 1 we need to know that it preserves our invariant. And indeed, just like here, it is a step in the correctness proof: if the procedure returns at least $\mathbf{RM}^w(i, j)$, then $S_{(i,j)}$ continuously reduces to the recognised language [Mur08]. The remaining cases essentially correspond to the items in Lemma 4 (below).

Lemma 4. *Assume that q is a state of $\mathcal{A}$, $\mathcal{B}$ is the SCC of $\mathcal{A}$ containing q, and p is a state of $\mathcal{A}$ reachable from q (from the same or different SCC).*

1. $\mathrm{L}(\mathcal{A}, p) \le_{\mathrm{W}} \mathrm{L}(\mathcal{A}, q)$.
2. $\mathrm{L}(\mathcal{A}^-, q) \le_{\mathrm{W}} \mathrm{L}(\mathcal{A}, q)$.
3. *If an accepting loop is reachable from q, then* $S_{(0,1)} \le_{\mathrm{W}} \mathrm{L}(\mathcal{A}, q)$.
4. *If a rejecting loop is reachable from q, then* $S_{(1,2)} \le_{\mathrm{W}} \mathrm{L}(\mathcal{A}, q)$.

5. *If p is $(\forall, 0)$-replicated by $\mathcal{B}$ then* $\left(\mathrm{L}(\mathcal{A}, p)\right)^{\forall} \leq_{\mathrm{W}} \mathrm{L}(\mathcal{A}, q)$.
6. *If p is $(\exists, 1)$-replicated by $\mathcal{B}$ then* $\left(\mathrm{L}(\mathcal{A}, p)\right)^{\exists} \leq_{\mathrm{W}} \mathrm{L}(\mathcal{A}, q)$.

Proof. The proof is based on the following observation. Let $t \in \mathrm{PTr}_A$ be a partial tree and $\rho = \rho(\mathcal{A}, t, q_I)$ be the run of an automaton $\mathcal{A}$ on t. We say that t *resolves $\mathcal{A}$ from $q_I \in Q^{\mathcal{A}}$* if $\rho(h) \neq *$ for each hole h of t and whenever $t\restriction_{vd}$ is the only total tree in $\{t\restriction_{v\mathtt{L}}, t\restriction_{v\mathtt{R}}\}$, either $\rho(vd) = *$ or $\mathbf{G}_\rho(\mathcal{A}, t\restriction_{vd}, \rho(vd))$ is losing for the owner of v. Assume that a tree t with a single hole h resolves $\mathcal{A}$ from q_I and take $\rho = \rho(\mathcal{A}, t, q_I)$. The notion of resolving is designed precisely so that $t[h := s] \in \mathrm{L}(\mathcal{A}, q_I)$ iff $s \in \mathrm{L}(\mathcal{A}, \rho(h))$ for all $s \in \mathrm{Tr}_A$.

Let us begin with (1). Since all the states of $\mathcal{A}$ are non-trivial, we can construct a tree t with a hole h such that t resolves $\mathcal{A}$ from q and the state $\rho(\mathcal{A}, t, q)(h)$ is p. In that case $t[h := s] \in \mathrm{L}(\mathcal{A}, q)$ if and only if $s \in \mathrm{L}(\mathcal{A}, p)$. Therefore, the function $s \mapsto t[h := s]$ is a continuous reduction witnessing that $\mathrm{L}(\mathcal{A}, p) \leq_{\mathrm{W}} \mathrm{L}(\mathcal{A}, q)$.

For (2), recall that $\mathcal{A}^-$ is obtained from $\mathcal{A}$ by turning some choices for $\forall$ to $\top$; that is, some transitions $\delta(q', a)$ of the form $(q_{\mathtt{L}}, \mathtt{L}) \wedge (q_{\mathtt{R}}, \mathtt{R})$ are set to $(q_{\mathtt{L}}, \mathtt{L})$, $(q_{\mathtt{R}}, \mathtt{R})$, or $\top$. This means that if a node v of tree t has label a and gets state q' in the associated run $\rho(\mathcal{A}^-, t, q)$, then $t\restriction_{v\mathtt{L}}$, $t\restriction_{v\mathtt{R}}$, or both of them, respectively, are immediately accepted by $\mathcal{A}^-$. In the corresponding run of the original automaton $\mathcal{A}$, however, these subtrees will be inspected by the players and we should make sure they are accepted. Since $q_{\mathtt{L}}$ and $q_{\mathtt{R}}$ are non-trivial in $\mathcal{A}$, we can do it by replacing these subtrees with $t_{q_{\mathtt{L}}} \in \mathrm{L}(\mathcal{A}, q_{\mathtt{L}})$, or $t_{q_{\mathtt{R}}} \in \mathrm{L}(\mathcal{A}, q_{\mathtt{R}})$, accordingly. This gives a continuous reduction of $\mathrm{L}(\mathcal{A}^-, q)$ to $\mathrm{L}(\mathcal{A}, q)$.

To prove (3), let us fix a state p on an accepting loop C, reachable from q. By (1) and transitivity of $\leq_{\mathrm{W}}$, it is enough to show that $S_{(0,1)} \leq_{\mathrm{W}} \mathrm{L}(\mathcal{A}, p)$. Let t be a tree with hole h such that t resolves $\mathcal{A}$ from p, the state $\rho(\mathcal{A}, t, p)$ is p, and the states on the shortest path from the root to h correspond to the accepting loop C. Since all states in $\mathcal{A}$ are non-trivial, we can also find a full tree $t' \notin \mathrm{L}(\mathcal{A}, p)$. Let $t_0 = t'$ and $t_n = t[h := t_{n-1}]$ for $n > 0$, and let t_∞ be the tree defined co-inductively as

$$t_\infty = t[h := t_\infty]\,.$$

Then, $t_n \notin \mathrm{L}(\mathcal{A}, p)$ for all $n \geq 0$, but $t_\infty \in \mathrm{L}(\mathcal{A}, p)$. To get a continuous function reducing $S_{(0,1)}$ to $\mathrm{L}(\mathcal{A}, p)$, map tree $s \in \mathrm{Tr}_{\{\bot,\top\}}$ to t_m, where $m = \min\left\{i \mid s(\mathtt{L}^i\mathtt{R}) = \bot\right\}$, or to t_∞ if $\left\{i \mid s(\mathtt{L}^i\mathtt{R}) = \bot\right\}$ is empty. Item (4) is analogous.

For (5), let us assume that $\delta(q, a) = (q_{\mathtt{L}}, \mathtt{L}) \wedge (p, \mathtt{R})$ is the transition witnessing that p is $(\forall, 0)$-replicated by $\mathcal{A}$. Let us also fix the path $q_{\mathtt{L}} \to q$ with minimal priority 0. Now, let t be a tree with a hole h that resolves $\mathcal{A}$ from q and the value of the run of $\mathcal{A}$ in h is q. Similarly, let t' be the tree with a hole h' that resolves $\mathcal{A}$ from $q_{\mathtt{L}}$ and the value of the respective run is q. Let us construct a continuous function that reduces $\left(\mathrm{L}(\mathcal{A}, p)\right)^{\forall}$ to $\mathrm{L}(\mathcal{A}, q)$. Assume that a given tree s has subtrees s_i under the nodes $\mathtt{L}^i\mathtt{R}$. Let us define co-inductively t_i as

$$t_i = a(t'[h' := t_{i+1}], s_i),$$

i.e. the tree with the root labelled by a and two subtrees: $t'[h' := t_{i+1}]$ and s_i. Finally, let $f(s)$ be $t[h := t_0]$. Note that the run $\rho(\mathcal{A}, f(s), q)$ labels the hole h of t by q'. Therefore, $f(t) \in \mathrm{L}(\mathcal{A}, q)$ if and only if $t_0 \in \mathrm{L}(\mathcal{A}, q')$ and $t_i \in \mathrm{L}(\mathcal{A}, q)$ if and only if $t_{i+1} \in \mathrm{L}(\mathcal{A}, q)$ and $s_i \in \mathrm{L}(\mathcal{A}, p)$. Since the minimal priority on the path from t_i to t_{i+1} is 0, if no s_i belongs to $\mathrm{L}(\mathcal{A}, p)$ then $f(t) \notin \mathrm{L}(\mathcal{A}, q)$. Therefore, f is in fact the desired reduction. The proof of (6) is entirely analogous. □

Using Lemma 4, and the guarantees for deterministic automata discussed earlier, we prove Lemma 2 as follows.

Proof of Lemma 2. By induction on the recursion depth of the algorithm execution we prove that if $\mathrm{wclass}(\mathcal{A}, p) \geq \mathbf{RM}^w(i, j)$ then $S_{(i,j)} \leq_{\mathrm{W}} \mathrm{L}(\mathcal{A}, p)$.

Let us start with the lowest level. Assume that $(i, j) = (0, 1)$ (for $(1, 2)$ the proof is analogous). Examining the algorithm we see that this is only possible if there is an accepting loop in $\mathcal{A}$, reachable from q. Then, by Lemma 4 Item (3), $S_{(0,1)} \leq_{\mathrm{W}} \mathrm{L}(\mathcal{A}, q)$.

For higher levels we proceed by case analysis. First we cover the possible reasons why equation (2) can give at least $\mathbf{RM}^w(i, j)$. If $\overline{\mathrm{wclass}(\bar{\mathcal{B}}, q)} \geq \mathbf{RM}^w(i, j)$, the invariant follows immediately from the guarantees for deterministic automata, and the duality between indices and between Skurczyński languages. If $\mathrm{wclass}(\mathcal{A}, p) \geq \mathbf{RM}^w(i, j)$ for some $p \in F$, we use the fact that $\mathrm{L}(\mathcal{A}, p) \leq_{\mathrm{W}} \mathrm{L}(\mathcal{A}, q)$, and get $S_{(i,j)} \leq_{\mathrm{W}} \mathrm{L}(\mathcal{A}, q)$ by transitivity. Then, assume that $\mathrm{wclass}(\mathcal{A}, p)^{\exists} \geq \mathbf{RM}^w(i, j)$ for some $p \in F_{\exists,1}$ (for $p \in F_{\forall,0}$ the proof is analogous). That means that $\mathrm{wclass}(\mathcal{A}, p) \geq \mathbf{RM}^w(0, j')$ such that $(\mathbf{RM}^w(0, j'))^{\exists} = \mathbf{RM}^w(1, j' + 2) \geq \mathbf{RM}^w(i, j)$. By the inductive hypothesis $S_{(0,j')} \leq_{\mathrm{W}} \mathrm{L}(\mathcal{A}, p)$, so by the monotonicity of $\exists$ and Lemma 4 Item (6), $S_{(1,j'+2)} = \big(S_{(0,j')}\big)^{\exists} \leq_{\mathrm{W}} (\mathrm{L}(\mathcal{A}, p))^{\exists} \leq_{\mathrm{W}} \mathrm{L}(\mathcal{A}, q)$. But since $\mathbf{RM}^w(1, j' + 2) \geq \mathbf{RM}^w(i, j)$, by the Wadge ordering of Skurczyński's languages $S_{(i,j)} \leq_{\mathrm{W}} S_{(1,j'+2)}$, and $S_{(i,j)} \leq_{\mathrm{W}} \mathrm{L}(\mathcal{A}, q)$ follows by transitivity.

Finally, assume that $\mathrm{wclass}(\mathcal{A}, q)$ is computed according to (3); that is, the component $\mathcal{B}$ contains an $\forall$-branching transition $\delta(q', a) = (q_{\mathtt{L}}, \mathtt{L}) \wedge (q_{\mathtt{R}}, \mathtt{R})$. As we have already observed, the hypothesis of the theorem implies that in that case $\max_{\Omega}(q_{\mathtt{L}} \to q') = 0$ and $\max_{\Omega}(q_{\mathtt{R}} \to q') \leq 1$ (or symmetrically). That means that $q_{\mathtt{R}}$ is $\forall, 0$-replicated by $\mathcal{B}$, so by Lemma 4 Item (5), $(\mathrm{L}(\mathcal{A}, q_{\mathtt{R}}))^{\forall} \leq_{\mathrm{W}} \mathrm{L}(\mathcal{A}, q)$. But since $\mathcal{B}$ is strongly connected, q is reachable from $q_{\mathtt{L}}$ and $q_{\mathtt{L}}$, so by Lemma 4 Item (1) we have $\mathrm{L}(\mathcal{A}, q) \leq_{\mathrm{W}} \mathrm{L}(\mathcal{A}, q_{\mathtt{R}})$. Since $\forall$ is monotone, we conclude that

$$(\mathrm{L}(\mathcal{A}, q))^{\forall} \leq_{\mathrm{W}} \mathrm{L}(\mathcal{A}, q)\,. \tag{4}$$

(It looks paradoxical, but note that $(L^{\forall})^{\forall} \leq_{\mathrm{W}} L^{\forall}$ for all L.) As $\mathrm{wclass}(\mathcal{A}, q) \geq \mathbf{RM}^w(i, j)$, it must hold that $\mathrm{wclass}(\mathcal{A}^{-}, q) \geq \mathbf{RM}^w(1, j')$ for some j' such that $(\mathbf{RM}^w(1, j'))^{\forall} = \mathbf{RM}^w(0, j') \geq \mathbf{RM}^w(i, j)$. By the induction hypothesis, $S_{(0,j')} \leq_{\mathrm{W}} \mathrm{L}(\mathcal{A}^{-}, q)$. Consequently, by Lemma 4 Item (2) and transitivity, $S_{(0,j')} \leq_{\mathrm{W}} \mathrm{L}(\mathcal{A}, q)$. Since the operation $\forall$ is monotone, (4) gives $S_{(0,j')} = \big(S_{(1,j')}\big)^{\forall} \leq_{\mathrm{W}} \mathrm{L}(\mathcal{A}, q)$, and we conclude by the $\leq_{\mathrm{W}}$ ordering of Skurczyński's languages. □