Definable Operations On Weakly Recognizable Sets of Trees

Jacques Duparc[1], Alessandro Facchini[2], and Filip Murlak[3]

1 **University of Lausanne, Switzerland**
`jacques.duparc@unil.ch`
2 **University of Warsaw, Poland**
`facchini@mimuw.edu.pl`
3 **University of Warsaw, Poland**
`fmurlak@mimuw.edu.pl`

Abstract

Alternating automata on infinite trees induce operations on languages which do not preserve natural equivalence relations, like having the same Mostowski–Rabin index, the same Borel rank, or being continuously reducible to each other (Wadge equivalence). In order to prevent this, alternation needs to be restricted to the choice of direction in the tree. For weak alternating automata with restricted alternation a small set of computable operations generates all definable operations, which implies that the Wadge degree of a given automaton is computable. The weak index and the Borel rank coincide, and are computable. An equivalent automaton of minimal index can be computed in polynomial time (if the productive states of the automaton are given).

1998 ACM Subject Classification F.4.3 Formal languages, F.4.1 Mathematical Logic

Keywords and phrases alternating automata, Wadge hierarchy, index hierarchy

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2011.363

1 Introduction

The structure of a regular language of infinite trees can be analyzed in terms of recognizing automata, defining formulas, or topological properties. Each approach defines a hierarchy of classes of similar languages: the Mostowski–Rabin index hierarchy, the μ-calculus alternation hierarchy, the Borel hierarchy, the Wadge hierarchy. Sometimes complementary, sometimes closely related, together they approximate the missing canonical representation of regular languages. Understanding them has been a goal pursued for decades, bringing spectacular successes like the Wagner hierarchy for regular languages of infinite words [24], providing a full characterization of the topological and combinatorial structure of a language in terms of certain patterns in the recognizing deterministic automaton. Various versions of this pattern method were successfully applied to deterministic automata on trees, resulting in a full classification in terms of Wadge equivalence [16, 18], non-deterministic index [19], and weak alternating index [17].

Owing to the elegant correspondence between certain set-theoretical and ordinal operations [4], the whole Wadge hierarchy of Borel sets of finite rank can be generated with several simple operations, starting from the empty set. The pattern method builds on this result. In order to obtain lower bounds for the Wadge hierarchy of the considered class of automata, it is often enough to check that some operations are definable within the class [5, 6].

In obtaining upper bounds and computability results, the pattern method relies on certain compositionality of deterministic automata with respect to the equivalence relations of having

31st Int'l Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2011).
Editors: Supratik Chakraborty, Amit Kumar; pp. 363–374
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

the same Mostowski–Rabin index, the same Borel rank, or being continuously reducible to each other (Wadge equivalence). In a deterministic automaton each sub-automaton can be replaced with any automaton recognizing an equivalent language without influencing the equivalence class of the whole language. More generally, each automaton can be seen as a result of an operation performed on sub-automata by means of some connecting automaton. If the connecting automaton is deterministic, the operation induces an operation on the equivalence classes of the corresponding languages (see [16, 18], and also [19]). Sometimes, these operations can be expressed in terms of computable ordinal operations on Wadge degrees, and the degree of the recognized language can be obtained by bottom up evaluation starting from the simple sub-automata [16].

For alternating automata this approach fails in general, because the set-theoretical operation of union, easily simulated within an alternating automaton, is not an operation on the equivalence classes. Indeed, the union of arbitrarily complicated languages can be the whole space. Does it mean that the pattern method is confined to deterministic automata? Recently it has been shown that the method can be extended beyond deterministic automata, but the class of considered languages was very small [7]. In this paper we introduce a large syntactic class of the automata inducing operations compatible with the Wadge equivalence—we call them *game automata*—and show that it is the largest such class satisfying natural closure conditions (Sect. 4). We then focus on weak automata, and identify a small set of operations on Wadge equivalence classes which generate all other definable operations (Sect. 5). Based on this we show how to compute the Wadge degree and the Borel rank of weak game automata (Sect. 6). Finally, we prove that the Borel rank and the weak index coincide for weak game automata, which leads to an algorithm computing the weak index, and constructing the equivalent automaton with minimal index (Sect. 7).

Due to space limitations many proofs are moved to the full version of the paper [8].

2 Alternating Tree Automata

Let T_Σ denote the set of *(full infinite binary) trees* over an alphabet Σ, i.e., functions $t : \{0,1\}^* \to \Sigma$. Given $v \in \mathsf{dom}(t)$, by $t.v$ we denote the subtree of t rooted in v.

A *alternating tree automaton* $\langle \Sigma, Q, q_I, \delta, \mathsf{rank} \rangle$ consisting of a finite alphabet Σ, a finite set of states Q, an initial state $q_I \in Q$, a transition function $\delta : Q \times \Sigma \to \mathcal{B}^+(\{0,1\} \times Q)$, where $\mathcal{B}^+(\{0,1\} \times Q)$ denotes the set of positive boolean formulae over $\{0,1\} \times Q$, and a rank function $\mathsf{rank} : Q \to \mathbb{N}$. As usual, A accepts $t \in T_\Sigma$ iff the player $\Diamond$ has a winning strategy in the induced max-parity game (see [8] for details). To underline this connection, we write transitions with $\square$ and $\diamond$ instead of $\wedge$ and $\vee$, e.g., $\delta(p,\sigma) = ((0,p) \square (0,q)) \diamond (1,q)$, or $p \xrightarrow{\sigma} ((0,p) \square (0,q)) \diamond (1,q)$. The class of all alternating automata is denoted by ATA.

An alternating tree automaton A is

- *weak* (WATA), if for all $q, q' \in Q$, if q is reachable from q' and q' is reachable from q, then $\mathsf{rank}(q) = \mathsf{rank}(q')$;
- *linear* (LATA), if for all $q, q' \in Q$, if q is reachable from q' and q' is reachable from q, then $q = q'$, and for each $q \in Q$ either all $\delta(q,\sigma)$ use only $\square$ or all use only $\diamond$;
- *deterministic* (DTA), if for all $q \in Q$, $\sigma \in \Sigma$, $\delta(q,\sigma) \in \{(0,p) \square (1,q) \mid p, q \in Q\}$.

A state q is *reachable* from p if there exists a *path* in A from p to q, i.e., a sequence of states and alphabet symbols $p_0\sigma_0p_1\sigma_1 \dots p_{k-1}\sigma_k p_k$ such that $p_0 = p$, $p_k = q$, and p_{i+1} occurs in $\delta(p_i, \sigma_i)$ for all $i < k$. Throughout the paper we assume that all states are reachable from the initial state. By convention, $\top$ is a singled out all-accepting state, and $\bot$ is an all-rejecting state. We assume that all other states are *non-trivial*, i.e., accept some tree and reject some tree. For every state q which is not the initial state q_I of the automaton A, by A_q we denote the automaton corresponding exactly to A except the fact that the initial state now is q and not q_I. We say that a state q of A is *productive* if $L(A_q) \neq \varnothing$.

The *(Mostowski–Rabin) index of an automaton* is given by $(i, j) \in \{0, 1\} \times \omega$, where i is the minimal and j is the maximal value of rank (scaling down the priorities we can always assume that the smallest rank is 0 or 1). Classes of languages recognizable with automata of index (i, j) form the so-called *index hierarchy*. By a result of Bradfield [3], we know that the index hierarchy for alternating tree automata, is strict. It is well-known that the class of weakly recognizable languages forms a strict hierarchy with respect to the index of the recognizing weak automata (cf. [1]). In the latter case we speak of the *weak index hierarchy*.

3 Borel classes and Wadge reductions

Consider the space T_Σ equipped with the standard Cantor (prefix) topology, that is the topology where a basic open set is the set all trees that extend a certain finite tree. Recall that the class of Borel sets of a topological space X is the closure of the class of open sets of X by countable unions and complementation. For a topological space X, the initial finite levels of the Borel hierarchy are defined as follows:

- $\mathbf{\Sigma}^0_1(X)$ is the class of open subsets of X,
- $\mathbf{\Pi}^0_n(X)$ contains complements of sets from $\mathbf{\Sigma}^0_n(X)$,
- $\mathbf{\Sigma}^0_{n+1}(X)$ contains countable unions of sets from $\mathbf{\Pi}^0_n(X)$.

By convention $\mathbf{\Sigma}^0_0(X) = \{\varnothing\}$ and $\mathbf{\Pi}^0_0(X) = \{X\}$.

The classes defined above are closed under inverse images of continuous functions. Let $\mathcal{C}$ be one of those classes. A set U is called $\mathcal{C}$-hard, if each set in $\mathcal{C}$ is an inverse image of U under some continuous function. If additionally $U \in \mathcal{C}$, U is said to be $\mathcal{C}$-complete. It is well known that every weakly recognizable tree language is a member of a Borel class of finite rank ([6, 14]). The rank of a language is the rank of the minimal Borel class the language belongs to. It can be seen as a coarse measure of complexity of languages.

A much finer measure of the topological complexity is the *Wadge degree*. If $T, U \subseteq T_\Sigma$, we say that T is *continuously (or Wadge) reducible* to U, $T \leqslant_W U$ in symbols, if there exists a continuous function f such that $T = f^{-1}(U)$. For a Borel class $\mathcal{C}$, T is $\mathcal{C}$-hard if $U \leqslant_W T$ for every $U \in \mathcal{C}$. We write $T \equiv_W U$ whenever $T \leqslant_W U \leqslant_W T$, and $T <_W U$, if $T \leqslant_W U$ but not $U \leqslant_W T$. The *Wadge hierarchy* is the partial order induced by $<_W$ on the $\equiv_W$-equivalence classes of Borel sets.

An alternative characterization of continuous reducibility can be given in terms of games. Let T and U be two arbitrary sets of trees. The *Wadge game* $\mathcal{W}(T, U)$ is played by two players, player I and player II. Each player builds a tree, say t_{I} and t_{II}, level by level. In every round, player I plays first, and both players add one level to their trees. Player II is allowed to skip her turn, but not forever. Player II wins the game if $t_{\mathrm{I}} \in T \Leftrightarrow t_{\mathrm{II}} \in U$.

▸ **Lemma 1** ([23]). *Let $T, U \subseteq T_\Sigma$. Then $T \leqslant_W U$ iff Player II has a winning strategy in the game $\mathcal{W}(T, U)$.*

An ordinal number is the order type of a well-ordered set. The least infinite ordinal is denoted by ω and corresponds to the order-type of the set of all natural numbers. We say that an ordinal α is countable if there is a bijection between α and ω. The first uncountable ordinal is denoted by ω_1. A subset B of an ordinal α is said to be *cofinal* if for every $a \in \alpha$ there exists some $b \in B$ such that $a \in b$. The *cofinality* of an ordinal α is thence the smallest ordinal β that is the order type of a cofinal subset of α.

Recall that a language L is called *self dual* if it is equivalent to its complement, otherwise it is called *non self dual.* From Borel determinacy [13], if $T, U \subseteq T_\Sigma$ are Borel, then $\mathcal{W}(T, U)$ is determined. As a consequence, a variant of Martin-Monk's result (cf. [11]) shows that $<_W$ is well-founded. Thus, we can associate to every Borel language an ordinal, called the *Wadge degree*, i.e. for sets of finite Borel rank, their Wadge degree is inductively defined by:

- $d_W(\varnothing) = d_W(\varnothing^\complement) = 1$,
- $d_W(L) = \sup\{d_W(K) + 1 \colon K \text{ non self dual}, K <_W L\}$ for $L >_W \varnothing$, non self-dual,
- $d_W(L) = \sup\{d_W(K) \colon K \text{ non self dual}, K <_W L\}$ for L self-dual.

For instance, open, non-closed sets have degree 2, just like closed, non-open sets. All clopens have degree 1. Let $\exp(\alpha) = \omega_1^\alpha$, and let ${}^{\omega_1}\epsilon_0 = \sup_{n \in \omega} \exp^n(\omega_1)$, the least fixpoint of the ordinal exponentiation of base ω_1. This is known to be the height of the Wadge hierarchy of all tree languages (recognizable or not) of finite Borel rank. More precisely, if L is $\mathbf{\Sigma}^0_n$-complete for $n > 1$, then $d_W(L) = \exp^{n-1}(1)$ for $n > 1$ (cf. [4]).

For each degree there are exactly three equivalence classes with the same degree, represented by U, $U^\complement$ and $U^\pm = \{t \mid t(\epsilon) = a, t.0 \in U\} \cup \{t \mid t(\epsilon) \neq a, t.0 \notin U\}$ for some non self-dual set U and $a \in \Sigma$. It easy to check that $U, U^\complement <_W U^\pm$ and $U^\pm$ is self-dual.

For each non self-dual set one can determine its sign, $+$ or $-$, which specifies precisely the $\equiv_W$-class [4]. For sets $U \subseteq T_\Sigma$ with $d_W(U)$ of countable cofinality, the sign is $+$ if U is Wadge equivalent to the set of trees over $\Sigma \cup \{c\}$, $c \notin \Sigma$, which have no c on the leftmost branch, or the first c is in the node 0^i and $t.0^i \in U$. The sign is $-$ if U is equivalent to the complement of this set. For instance, $\varnothing$ and open, non-closed sets have sign -, while the whole space and closed, non-open sets have sign $+$. For sets of cofinality ω_1, the definition is more complicated, but $\mathbf{\Sigma}^0_n$-complete sets have sign $-$, and $\mathbf{\Pi}^0_n$-complete sets have sign $+$. All self-dual sets by definition have sign $\pm$. Thus an ordinal $\alpha < {}^{\omega_1}\epsilon_0$ and a sign $\epsilon \in \{+, -, \pm\}$, determine a $\equiv_W$-class, denoted $[\alpha]^\epsilon$.

4 Game automata

For A, B (over the same alphabet) and an occurrence of a state q in a transition $\delta(p, \sigma)$ of A, the substitution A_B is obtained by replacing the occurrence of q in $\delta(p, \sigma)$ with the initial state of B. The mapping $B \mapsto A_B$ induces an operation on recognized languages, but it need not preserve coarser equivalence relations, like Wadge equivalence.

As pointed out in the introduction, the operation of union is not compatible with such equivalence relations. The same is true of intersection.

▸ **Example 2.** Take $\Sigma = \{0, 1, 2\}$ and consider $(\Sigma^*(1+2))^\omega$ and $(\Sigma^*2)^\omega$. Clearly, $(\Sigma^*(1+2))^\omega \leqslant_W (\Sigma^*2)^\omega$ as witnessed by the letter-to-letter morphism $0 \mapsto 0$ and $1, 2 \mapsto 2$. The converse reduction is given by the inclusion. Taking union with Σ^*0^ω, we obtain $(\Sigma^*(1+2))^\omega \cup \Sigma^*0^\omega = \Sigma^\omega$, and $(\Sigma^*2)^\omega \cup \Sigma^*0^\omega \not\equiv_W \Sigma^\omega$. The language $(\Sigma^*2)^\omega \cup \Sigma^*0^\omega$ is at the level $\mathbf{\Delta}^0_3$ of the Borel hierarchy, a deterministic automaton requires three ranks to recognize it, and an alternating automaton needs two. This makes it much more complex than the whole space Σ^ω, which can be recognized by a deterministic automaton with a single state, whose rank is 0. Similarly, intersecting with $\Sigma^*(0+1)^\omega$ we obtain $\Sigma^*(0^*1)^\omega$, and the empty set, which have very different complexity.

In order to ensure that substitution is well-behaved, we need to prevent the automata from simulating union and intersection. We call a transition $\delta(q, a)$ *ambiguous* if it contains two occurrences of some direction $d \in \{0, 1\}$.

▸ **Fact 3.** Let $\mathcal{C} \subseteq$ ATA be a class of automata over a fixed alphabet with at least two letters, closed under substitution and containing the one-state all-rejecting and all-accepting automata. Substitution preserves the Wadge equivalence in $\mathcal{C}$ iff no automaton of $\mathcal{C}$ has an ambiguous transition.

Proof. Assume for simplicity that the alphabet contains the symbols $0, 1, 2$. Starting from the all-accepting and all-rejecting automata over the alphabet $\{0, 1, 2\}$ we can obtain automata $A, A^{\complement}B, B^{\complement}$ recognizing languages L_{0^ω}, $(L_{0^\omega})^{\complement}$, $L_{(0+1)^\omega}$, $(L_{(0+1)^\omega})^{\complement}$ respectively, where L_α stands for the set of trees whose leftmost branch is a word from the language defined by the expression α. Observe that $L(A) \equiv_W L(B)$, but $L(A) \cup L(B^{\complement}) \not\equiv_w L(B) \cup L(B^{\complement})$ and $L(A) \cap L(A^{\complement}) \not\equiv_w L(B) \cap L(A^{\complement})$.

Let $C \in \mathcal{C}$ and let $q_0\sigma_0 q_1\sigma_1 \dots q_k$ be path from the initial state q_0 to a state q_k such that for some σ_k, $\delta(q_k, \sigma_k)$ is an ambiguous transition. By substituting the all-accepting and all-rejecting automata, we can assume that $\delta(q_i, \sigma_i) = (d_i, q_{i+1})$ for $i < k$ and $\delta(q_k, \sigma_k) = (d_k, p_0) \diamond (d_k, p_1)$ or $\delta(q_k, \sigma_k) = (d_k, p_0) \,\Box\, (d_k, p_1)$ for some states p_0, p_1. Assume that $\delta(q_k, \sigma_k) = (d_k, p_0) \diamond (d_k, p_1)$, and let C' be the result of replacing the occurrence of p_0 with the initial state of B, and the occurrence of p_1 with the initial state of $B^{\complement}$. For C'_A, obtained by replacing the initial state of B with the initial state of A, we have $L(C'_A) \equiv_W L(A) \cup L(B^{\complement})$, and $L(C') \equiv_W L(B) \cup L(B^{\complement})$, which concludes the proof. For $\delta(q_k, \sigma_k) = (d_k, p_0) \,\Box\, (d_k, p_1)$, use $A^{\complement}$ instead of $B^{\complement}$. ◂

Observe that each non-ambiguous transition has one of the four forms: $(0, p)$, $(1, p)$, $(0, p) \diamond (1, q)$, or $(0, p) \,\Box\, (1, q)$.

▸ **Definition 4.** A *game automaton* (GA) is an alternating automaton without ambiguous transitions. For notational simplicity, we assume that

$$\delta\colon Q \times \Sigma \to \{p \diamond q \mid p, q \in Q \backslash \{\top\}\} \cup \{p \,\Box\, q \mid p, q \in Q \backslash \{\bot\}\} ,$$

where $p \diamond q$ and $p \,\Box\, q$ is interpreted as $(0, p) \diamond (1, q)$ and $(0, p) \,\Box\, (1, q)$, respectively.

A *weak game automaton* (WGA), is a game automaton which is also weak, and a *linear game automaton* (LGA) [7], is a game automaton which is linear.

Fact 3 implies that GA it the largest nontrivial subclass of ATA closed under substitution for which substitution preserves Wadge equivalence, and similarly for WGA $\subseteq$ WATA. In fact, a more general property holds for GA.

▸ **Fact 5.** For every GA A, B, B', every state q of A, and A_B, $A_{B'}$ obtained by replacing an occurrence of q with the initial state of B and B' respectively, it holds that

1. $L(B) \leqslant_W L(B')$ implies $L(A_B) \leqslant_W L(A_{B'})$,
2. $L(A_q) \leqslant_W L(A)$.

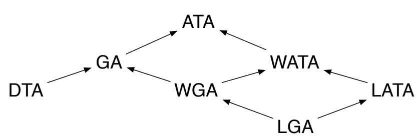

Figure 1

Relations between the classes are shown in Fig. 1 with arrows standing for class inclusion. The classes GA, WGA, and LGA are closed under complementation: the usual complementation procedure of increasing the ranks by one and swapping existential and universal transitions works. However they are neither closed under union nor intersection. For instance, let $L_\sigma = \{t \in T_{\{a,b\}} : t(0) = t(1) = \sigma\}$. Obviously, L_a and L_b are LGA-recognizable, but $L_a \cup L_b$ is not even GA-recognizable. Note that the last example also shows that all the inclusions in the diagram above are strict.

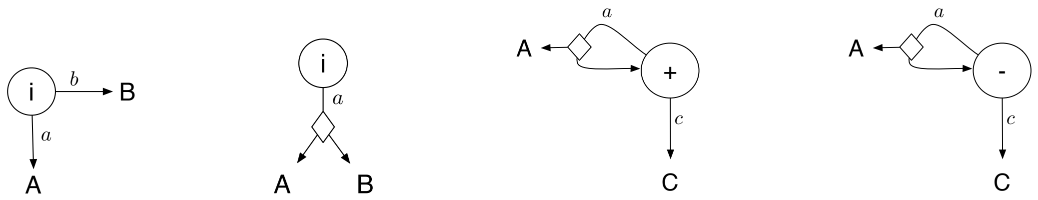

Figure 2 Automata constructions for $\sqcup$, $\diamond$, loop^+, $\exists$.

5 Operations induced by automata

LGA, investigated in [7], can be classified in terms of several simple set theoretic operations (we assume that the alphabet contains letters a, b, c):

$$\begin{aligned}
L \sqcup M &= \{t \mid t(\varepsilon) = a, t.0 \in L\} \cup \{t \mid t(\varepsilon) \neq a, t.0 \in M\} \ , \\
L \,\square\, M &= \{t \mid t.0 \in L \wedge t.1 \in M\} \ , \\
L \diamond M &= \{t \mid t.0 \in L \vee t.1 \in M\} \ , \\
\mathsf{loop}^-(L, M) &= \bigcup_{n\in\mathbb{N}} \{t \mid \text{first } c \text{ is in } 0^n, t.0^{n+1} \in M, \text{ and } t.0^\ell 1 \in L \text{ for all } \ell < n\} \ , \\
\mathsf{loop}^+(L, M) &= \bigcup_{n\in\mathbb{N}} \{t \mid \text{first } c \text{ is in } 0^n, \text{ and } t.0^{n+1} \in M \text{ or } t.0^\ell 1 \in L \text{ for some } \ell < n\} \cup \\
&\quad \cup \{t \mid t.(0^n) \neq c \text{ for all } n\} \, , \\
\forall(L, M) &= \mathsf{loop}^-(L, M) \cup \{t \mid t.(0^n) \neq c \text{ for all } n, \text{ and } t.0^\ell 1 \in L \text{ for all } \ell\} \, , \\
\exists(L, M) &= \mathsf{loop}^+(L, M) \cup \{t \mid t.(0^n) \neq c \text{ for all } n, \text{ and } t.0^\ell 1 \in L \text{ for some } \ell\} \, ,
\end{aligned}$$

where "first c is in 0^n" means that $t(0^n) = c$ and $t(0^k) \neq c$ for all $k < n$. Observe that $(L \,\square\, M)^\complement = L^\complement \diamond M^\complement$, $(\mathsf{loop}^+(L, M))^\complement = \mathsf{loop}^-(L^\complement, M^\complement)$, and $(\forall(L, M))^\complement = \exists(L^\complement, M^\complement)$.

These operations are definable by LGA: automata realizations for $\sqcup$, $\diamond$, loop^+, $\exists$ are shown in Fig. 2, and for $\square$, loop^-, $\forall$ they are obtained by replacing $\diamond$ with $\square$ and swapping the rank parities. Like all operations induced by GA, they are compatible with Wadge equivalence.

▸ **Fact 6.** Let op be one of the operations $\sqcup$, $\diamond$, loop^+, $\exists$, or their duals. Whenever $L \equiv_W L'$ and $M \equiv_W M'$, it holds that $\mathsf{op}(L, M) \equiv_W \mathsf{op}(L', M')$.

Up to Wadge equivalence, these operations are everything LGA are able to express.

▸ **Fact 7** ([7])**.** Up to Wadge equivalence, the closure of $\{\top, \bot\}$ under $\sqcup$, $\diamond$, loop^+, $\exists$, and their duals (or equivalently, complementation) gives exactly the family of sets recognized by LGAs. Moreover, for each LGA one can compute an equivalent canonical term over these operations and $\bot$, $\top$.

Since the operations preserve Wadge equivalence, they can be defined in terms of ordinal arithmetics and signs [4, 7]. For some operations the definitions are very simple, for instance

$$[\gamma_1]^{\epsilon_1} \sqcup [\gamma_2]^{\epsilon_2} = [\max(\gamma_1, \gamma_2)]^\epsilon, \text{ where } \epsilon = \begin{cases} \epsilon_1 & \text{if } \gamma_1 > \gamma_2 \\ \pm & \text{if } \gamma_1 = \gamma_2 \text{ and } \epsilon_1 \neq \epsilon_2, \\ \epsilon_2 & \text{otherwise} \end{cases}$$

$$\mathsf{loop}^+([\gamma]^\epsilon, [1]^-) = \left[\sup_k d_W(([\gamma]^\epsilon)^{\langle k\rangle})\right]^+ , \text{ where } U^{\langle k\rangle} = \underbrace{U \diamond U \diamond \cdots \diamond U}_{k}$$

$$\exists([\gamma]^\epsilon, [1]^-) = [\exp^{i+1} 1]^-, \text{ for } [\exp^i 1]^+ \leqslant_W [\gamma]^\epsilon \leqslant_W [\exp^{i+1} 1]^- \, .$$

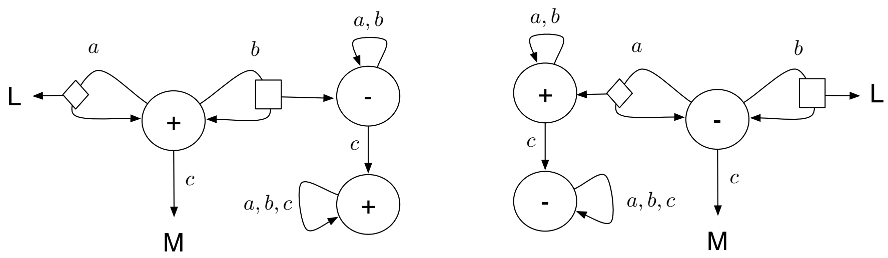

Figure 3 Operations definable with WGA.

Observe that the second equation, and its dual, imply that for all k

$$\mathsf{loop}^+(L, M) \geqslant_W L^{\langle k\rangle}\,, \qquad \mathsf{loop}^-(L, M) \geqslant_W L^{[k]}\,,$$

where $U^{[k]} = \underbrace{U \,\square\, U \,\square \cdots \square\, U}_{k}$. For $\diamond$ the ordinal definition has only been given for $\equiv_W$-classes inhabited by LGA-recognizable languages, $[\Phi] = \{[\alpha]^\epsilon \mid \alpha \in \Phi, \epsilon \in \{+, -, \pm\}\}$ with Φ denoting the set of ordinals of the form $\sum_{n=N}^{0} \beta_n + \alpha$ where $\alpha < \omega$ and each β_n is of the form $\exp^n(\omega)\eta + \sum_{p=P}^{1} \exp^n(p)k_p$ for some $\eta < \omega^\omega$ and $k_p < \omega$. Closure of $[\Phi]$ under $\sqcup$, $\diamond$, loop^+, $\exists$ (and their duals) was the technical core of the proof of Fact 7.

In this work we want to move to sets recognizable by WGA. Surprisingly, only two really new operations are introduced, $\mathsf{loop\text{-}reset}^+(L, M)$ and $\mathsf{loop\text{-}reset}^-(L, M)$. The automata constructions for them are shown in Fig. 3.

By a Wadge game argument we get a simple characterization in terms of ordinal arithmetics, showing that WGA-definable operations can multiply some Wadge degrees by ω_1.

▸ **Theorem 8.** *For every Wadge equivalence class $[\gamma]^\epsilon$ of a Borel language and $\mu \in \{+, -\}$*

$$\mathsf{loop\text{-}reset}^\mu([\gamma]^\epsilon, [1]^{\overline{\mu}}) = \begin{cases} [3]^{\overline{\mu}} & \text{if } [\gamma]^\epsilon \equiv_W [1]^\mu, \\ \left[d_W(\mathsf{loop}^+([\gamma]^\epsilon, [1]^-))\omega_1\right]^\mu & \text{otherwise}, \end{cases}$$

where $\overline{\mu} = +$ if $\mu = -$, $\overline{\mu} = -$ otherwise.

This operation is the source of difference between LGA and WGA, and allows WGA to inhabit much many Wadge equivalence classes than LGA. Thus, in our algorithm for WGA we use effective closure for a larger set of ordinals. Let Ω be the set of ordinals of the form $\Sigma_{i=K}^{0} \exp(\alpha_i)\eta_i$ where $\alpha_K, \alpha_{K-1}, \ldots, \alpha_0$ is a strictly decreasing sequence of ordinals from Φ, and $\eta_i < \omega$ for $\mathrm{cof}\,\alpha_i = \omega_1$ or $\mathrm{cof}\,\alpha_i < \omega$, and $\eta_i < \omega^\omega$ for $\mathrm{cof}\,\alpha_i = \omega$.

▸ **Lemma 9.** *$[\Omega]$ is closed under the operations $\sqcup$, loop^+, $\mathsf{loop\text{-}reset}^+$, $\exists$ (and their duals) and the result of the operation can be computed effectively.*

The proof is by induction, with the base cases covered by the closure property for $[\Phi]$.

6 Computing the Wadge degrees of WGA

For game automata, a run (computation tree) over an input tree t is a labeling of the input tree with states and modes ($\square$ or $\diamond$), induced by the transition function of the automaton. A transition taken from a node v determines the mode of v and the states in its children as follows: the root is labelled with the initial state, and if a node with label σ gets state q and

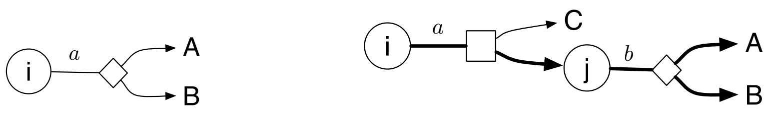

Figure 4 A simulation (the rank j must not be greater then i).

$q \xrightarrow{\sigma} q' \circ q''$ then v gets the mode $\circ$, and the left and right children get the states q' and q'' respectively. A run ρ is *resolved up to* a subtree ρ' if for all $v, v0, v1 \in \mathsf{dom}\,\rho$ such that exactly one node vd belongs to $\mathsf{dom}\,\rho'$, and for the remaining node vd' the sub-run $\rho.vd'$ is accepting if v's mode is $\Box$ and rejecting if it is $\Diamond$.

▸ **Definition 10.** A *simulation* of a run ρ in a run σ is a partial function $\eta : \mathsf{dom}\,\rho \to \mathsf{dom}\,\sigma$ such that

- $\mathsf{dom}\,\eta$ is a prefix closed subset of $\mathsf{dom}\,\rho$ (possibly with leaves and infinite branches);
- σ is resolved up to the subtree induced by the image of η;
- for each $v0, v1 \in \mathsf{dom}\,\eta$, $\eta(v0), \eta(v1)$ are descendants of $\eta(v)$, their closest common ancestor has the same mode as v , and the highest rank on the path from $\eta(v)$ to $\eta(vd)$ is equal to the rank of state in vd for $d = 0, 1$;
- for each leaf $v \in \mathsf{dom}\,\eta$, $\rho.v$ is accepting iff $\sigma.\eta(v)$ is accepting.

▸ **Lemma 11.** *If there is a game simulation of ρ in σ, then ρ is accepting iff σ is accepting.*

Proof. Each strategy in the parity game on ρ can be carried over to σ, and *vice versa.* ◂

▸ **Definition 12.** A *simulation* of an automaton A in an automaton B consists of a partition of Q^A into sets Q_1, Q_2, Q_3 and function $\eta\colon Q_1 \cup Q_2 \to Q^B$ such that

- $q_I^A \in Q_1$ and each transition of A originating in Q_1 leads to $Q_1 \cup Q_2$;
- whenever $q \xrightarrow{\sigma}_A q_0 \circ q_1$ for some $q \in Q_1$ and $\circ \in \{\Diamond, \Box\}$, there exist a path π from $\eta(q)$ to some p and paths π_i from some p_i to $\eta(p_i)$ for $i = 0, 1$ such that $p \xrightarrow{\tau}_B p_0 \circ p_1$ or $p \xrightarrow{\tau}_B p_1 \circ p_0$ and the highest rank on $\pi\tau\pi_i$ is equal to $\mathsf{rank}\,q_i$;
- for all $q \in Q_2$, $L(A_q) \leqslant_W L(B_{\eta(q)})$.

An example of a simulation is given in Fig. 4. A simulation of A in B immediately provides a continuous reduction from the set of accepting runs of A to the set of accepting runs of B. The next lemma follows by noticing that for GAs the set of accepting runs is Wadge equivalent to the recognized language.

▸ **Lemma 13.** *If there exists a simulation of A in B, then $L(A) \leqslant_W L(B)$.*

Strongly connected components (SCCs) of automata are defined as for graphs in terms of reachability. An SCC is *trivial* if it does not contain any loop. A transition $q \xrightarrow{\sigma} q' \circ q''$ is called *branching* if q, q', q'' belong to the same SCC.

▸ **Lemma 14.** *For each* WGA *one can effectively compute a Wadge equivalent* WGA *over $\{a, b, c\}$ without non-trivial loops.*

Proof. First we construct an automaton over a larger alphabet. We collapse each strongly connected component into one state, proceeding by induction on the DAG of SCCs. Let X be the root SCC, i.e., the SCC containing the initial state q_I. By induction hypothesis, we can assume that all other SCCs consist of a single state.

If there is a branching $\Box$-transition in X, set $q_I \xrightarrow{a} q_I \Box q_I$. Otherwise, set $q_I \xrightarrow{a_p} q_I \Box p$ for all $p \notin X$ such that $q \xrightarrow{\sigma} q' \Box p$ or $q \xrightarrow{\sigma} p \Box q'$ for some $q, q' \in X$. Define the transitions via b and b_p analogously, replacing $\Box$ with $\Diamond$. Finally, let $q_I \xrightarrow{c_{p \circ p'}} p \circ p'$ where $p \circ p'$ ranges over

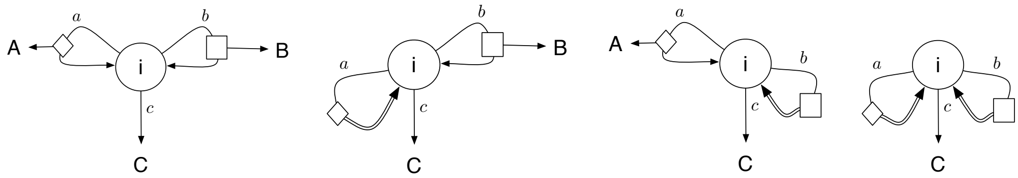

Figure 5 Strongly connected components of WGA over $\{a, b, c\}$ without non-trivial loops.

- $p \circ p'$ such that $p, p' \notin X$ and $q \xrightarrow{\sigma} p \circ p'$ for some $q \in X$, $\circ \in \{\diamond, \square\}$;
- $p \square \top$ such that $p \notin X$ and $q \xrightarrow{\sigma} q' \square p$ or $q \xrightarrow{\sigma} p \square q'$ for some $q, q' \in X$; and
- $p \diamond \bot$ such that $p \notin X$ and $q \xrightarrow{\sigma} q' \diamond p$ or $q \xrightarrow{\sigma} p \diamond q'$ for some $q, q' \in X$.

For each state q of the new automaton, extend $\delta(q, \sigma)$ to all symbols in the alphabet by using one of already defined transitions.

The original automaton can be simulated in the modified one by taking $Q_1 = X$, $Q_2 = \{p \mid q \xrightarrow{\sigma} p \circ r \text{ or } q \xrightarrow{\sigma} r \circ p \text{ for some } q \in X, r \in Q\}$, and $\eta(q) = q_I$ for $q \in Q_1$, and $\eta(p) = p$ for $p \in Q_2$. For the converse simulation, only change Q_1 to $\{q_I\}$, and for Q_2 and η keep the definitions above.

To reduce the alphabet to $\{a, b, c\}$, modify the construction as follows. In the case where there is no branching $\square$-transition in X, add a single transition $q_I \xrightarrow{a} q_I \square q_a$, where q_a is the initial state of the automaton recognizing $L(A_{p_1}) \sqcup L(A_{p_2}) \sqcup \dots \sqcup L(A_{p_k})$, where $\{p_1, p_2, \dots, p_k\}$ is the set over which p ranges in the original construction. For b the modification is analogous, and for c add $q_I \xrightarrow{\sigma} \bot \diamond q_c$, where q_c is the initial state of the automaton recognizing $\left[L(A_{p_1}) \circ_1 L(A_{p'_1})\right] \sqcup \left[L(A_{p_2}) \circ_2 L(A_{p'_2})\right] \sqcup \dots \sqcup \left[L(A_{p_k}) \circ_\ell L(A_{p'_\ell})\right]$, where $p_1 \circ_1 p'_1, p_2 \circ_2 p'_2, \dots, p_\ell \circ_\ell p'_\ell$ are the triples over which $p \circ p'$ ranges in the original construction. Observe that these modifications do not influence the Wadge equivalence class of the recognized language. ◂

Thus we can assume that each non-trivial SCC of a given WGA is of one of the four forms presented in Fig. 5. By a Wadge game argument we can further simplify the automaton.

▸ **Lemma 15.** *For each* WGA *one can compute effectively a Wadge equivalent* WGA *over* $\{a, b, c\}$ *without non-trivial loops and branching transitions (except those for* $\top$, $\bot$*).*

After these simplifications we apply Lemma 9 to compute the Wadge degrees.

▸ **Theorem 16.** *For a given* WGA *one can effectively compute the Wadge equivalence class of the language it recognizes.*

Proof. By Lemma 15, we can assume that the automaton is over $\{a, b, c\}$, has no non-trivial loops, and no branching transitions. By induction on the DAG of SCCs we prove that the Wadge equivalence class of the recognized language is in $[\Omega]$ and can be computed effectively.

If the whole automaton consists of a single SCC, the result is $[1]^+$ or $[1]^-$ depending on the rank of the unique state.

To perform the inductive step, it suffices to express the recognized language in terms of the operations from Lemma 9. If there is no transition from the initial state q_I that leads back to q_I, the recognized language can be presented as $\left[L(A_{p_a}) \circ_a L(A_{p'_a})\right] \sqcup \left[L(A_{p_b}) \circ_b L(A_{p'_b})\right] \sqcup \left[L(A_{p_c}) \circ_c L(A_{p'_c})\right]$, where $q_I \xrightarrow{\sigma} p_\sigma \circ_\sigma p'_\sigma$ for $\sigma = a, b, c$.

For the rest of the proof we assume that the automaton is of the form shown in the leftmost part of Fig. 5; we use the notation $q_I(A, B, C)$. For the remaining possibilities the computations are analogous. If the rank q_I is even, consider the following cases.

1. $L(A) \geqslant_W \forall(L(B), \top)$. Then $L(A) >_W L(B)^{[n]}$ for every $n < \omega$, and we have that either $L(q_I(A, B, C)) \equiv_W L(q_I(A, B', C))$ for some B' recognizing a $\mathbf{\Sigma}^0_1$-complete language, if $d_W(L(B)) \geqslant [2]^-$, or $L(q(A, B, C)) \equiv_W L(q_I(A, \top, C))$ otherwise. In the former case the recognized language is Wadge equivalent to $\mathsf{loop\text{-}reset}^+(L(A), L(C))$, in the latter case it is Wadge equivalent to $\mathsf{loop}^+(L(A), L(C))$.
2. $L(A) <_W L(\forall(B, \top))$. The recognized language is Wadge equivalent to $L(q_I(\bot, B, C))$, which gives $\forall(L(B), L(C))$.
3. $L(A) \equiv_W L(\forall(B, \top))^\complement$. In this case, as $L(A) >_W L(B)^{[n]}$ for every $n < \omega$, we conclude that the recognized language is Wadge equivalent to $L(A) \diamond L(q(\bot, B, C)) \equiv_W L(A) \diamond \forall(L(B), L(C))$.

For $\mathsf{rank}\, q_I$ odd, dualize the above argument. ◂

7 Borel rank and weak index

As an immediate corollary of Theorem 16 we obtain decidability of the Borel rank problem.

▸ **Corollary 17.** *The problem of deciding the Borel rank of a* WGA*-recognizable language is decidable.*

We will now proceed to prove that the weak index conjecture holds for languages recognized by WGA. It has long been known that one implication holds.

▸ **Proposition 18** ([14])**.** Let $A \in$ WGA with index $(0, n)$ (resp. $(1, n+1)$). Then it holds that $L(A) \in \Pi^0_n$ (resp. $L(A) \in \Sigma^0_n$).

Using the connections between the structure and topological complexity of automata explained in the previous sections, we can prove the converse for WGA.

▸ **Theorem 19.** *For languages recognizable by* WGA*, the Borel hierarchy and the weak index hierarchy coincide.*

Proof. By duality and Proposition 18 it suffices to show that each WGA A recognizing a $\mathbf{\Pi}^0_n$ language admits an equivalent WATA of index $(0, n)$. We proceed by induction on the DAG of SCCs of the automaton.

If $n = 0$, A accepts every tree, so it is equivalent to a single state automaton of index $(0, 0)$. If $n = 1$, A cannot contain a productive state reachable from a nontrivial rejecting SCC, so an equivalent $(0, 1)$ automaton can be obtained by setting the rank of all states reachable from non-trivial rejecting SCCs to 1 and the rank of the remaining states to 0.

Suppose that $n \geqslant 2$, and let X be the root SCC. If X has rank 0 (we can change it to 0 if X is trivial), by Fact 5 (2) and the induction hypothesis we can present all A_q with $q \notin X$ as $(0, n)$ automata and the claim follows.

Suppose X is non-trivial and has rank 1. Assume that X contains a branching $\diamond$-transition. Then it follows that for all states q, $L(A_q)$ is in $\mathbf{\Sigma}^0_{n-1}$ (otherwise, the whole language would be $\mathbf{\Sigma}^0_n$ hard). In consequence, for all states $p \notin X$, A_p can be transformed into an equivalent WATA of index $(1, n)$, and we conclude like before.

The remaining case is that of non-trivial X of rank 1, without branching $\diamond$-transitions. Observe that in this case, there are two reasons why a tree can be rejecting:

1. a path of the computation stays forever in X, and for all $\diamond$ transitions in this path, the branches leaving X are rejecting;
2. a rejecting path exits X, and for all $\diamond$ transitions in this path, branches leaving X are rejecting.

By induction hypothesis, all A_p can be transformed to WATA of index $(1,n)$ if $q \xrightarrow{\sigma} p \diamond q'$ or $q \xrightarrow{\sigma} q' \diamond p$ for some $q, q' \in X$, or $(0,n)$ otherwise. To check that the second condition does not hold, use A with the rank X changed to 0. For the first condition, use A' obtained from A by replacing $q \xrightarrow{\sigma} p \circ p'$ with $q \xrightarrow{\sigma} \bot \diamond \bot$, $q \xrightarrow{\sigma} p \sqcap q'$ with $q \xrightarrow{\sigma} \top \diamond q'$, and $q \xrightarrow{\sigma} q' \sqcap p'$ with $q \xrightarrow{\sigma} q' \diamond \top$ for all $q, q' \in X$, $p, p' \notin X$. The ϵ-transition introduced to implement conjunction can be removed by unraveling the first step of the computation, without changing the ranks. ◂

This way the weak index problem reduces to the Borel rank problem. The construction above in fact gives an effective way of constructing the equivalent WATA of minimal index.

▸ **Corollary 20.** *The problem of calculating the exact position in the weak index hierarchy of a language recognized by a* WGA *is decidable and an equivalent* WATA *can be constructed effectively (in polynomial time if the productive states are given).*

8 Conclusions

We have isolated the class of game automata, a wide class of automata inducing operations on Wadge equivalence classes. For *weak* game automata we were able to use this property to describe all definable operations in terms of a small set of generators, and based on this we gave a procedure calculating the Wadge equivalence class of the language recognized by any given automaton. Using the structural information provided by the latter result we proved that the weak index hierarchy and the Borel hierarchy coincide, and gave algorithms computing the weak index and constructing an equivalent weak alternating automaton of the minimal index.

The results on the Wadge hierarchy subscribe to the line of reaserch aimed at investigating the hierarchies for families of languages recognized by various devices (cf. [5, 9, 21]). Usually, lower bounds on the heights of the hierarchies are easier to obtain, tight upper bounds are more difficult, and decidability results are scarce [7, 16, 24]. The peculiarity of this work is that we obtain computability of the Wadge degree without determining explicitly the inhabited levels of the hierarchy. Some lower bounds are easy to obtain based on our description of the induced operations and an upper bound is given by [Ω], but giving a full characterization of the inhabited levels seems to be a nontrivial task.

The class of automata we are considering has limited expressivity, but it seems to capture many interesting topological phenomena. Even more so in the unrestricted case, as game automata recognize the game languages recently considered by Arnold and Niwiński [2] in their study of the Wadge hierarchy of non-Borel regular languages. Currently, we are trying to drop the weakness restriction. One of the challenges is that for non-Borel languages the shape of Wadge hierarchy is unknown.

Despite the positive results concerning the hierarchy problems for weak game automata, and hopefully for non-weak, from the methodological point of view the message of this work is that we are reaching the limits of the topological approach to index problems. Pushing decidability results beyond game automata seems to require new techniques.

Acknowledgements

The second author is supported by a grant from the SNFS, n. PBLAP2-132006, while the third author is supported by the Polish government grant no. N N206 567840.

References

1 A. Arnold, J. Duparc, F. Murlak, D. Niwiński. On the Topological Complexity of Tree Languages. In J. Flum et al. (Eds.) *Logic and Automata - History and Perspectives*, Texts in Logic and Games, Amsterdam University Press: 9–28 (2007).

2 A. Arnold, D. Niwiński. Continuous Separation of Game Languages. *Fund. Info.*, 81(1–3): 19–28 (2008).

3 J. Bradfield. The Modal μ-Calculus Alternation Hierarchy is Strict. *Theor. Comput. Sci.* 195(2): 133–153 (1998).

4 J. Duparc. Wadge Hierarchy and Veblen Hierarchy Part 1: Borel Sets of Finite Rank. *J. Symb. Log.* 66(1): 56–86 (2001).

5 J. Duparc. A Hierarchy of Deterministic Context-Free ω-Languages. *Theoret. Comput. Sci.* 290:1253–1300 (2003).

6 J. Duparc, F. Murlak. On the Topological Complexity of Weakly Recognizable Tree Languages. *FCT*: 261–273 (2007).

7 J. Duparc, A. Facchini, F. Murlak. Linear Game Automata: Decidable Hierarchy Problems for Stripped-Down Alternating Tree Automata. *CSL*: 225–239 (2009).

8 J. Duparc, A. Facchini, F. Murlak. Definable Operations On Weakly Recognizable Sets of Trees. `http://www.mimuw.edu.pl/~fmurlak/papers/gamafull.pdf`.

9 O. Finkel. Borel Ranks and Wadge Degrees of ω-Context Free Languages. *Mathematical Structures in Computer Science* 16: 813–840 (2006).

10 S. Hummel, H. Michalewski, D. Niwiński. On the Borel Inseparability of Game Tree Languages. *STACS*: 565–576 (2009).

11 A. S. Kechris. *Classical Descriptive Set Theory.* Graduate Texts in Mathematics, Vol. 156. Springer-Verlag, New York (1995).

12 L. H. Landweber. Decision Problems for ω-Automata. *Math. Systems Theory* 3: 376–384 (1969).

13 D. A. Martin. Borel Determinacy. *Ann. of Math.* (2), 102(2): 363–371 (1975).

14 A. W. Mostowski. Hierarchies of Weak Automata and Weak Monadic Formulas. *Theoret. Comput. Sci.* 83: 323–335 (1991).

15 F. Murlak. On Deciding Topological Classes of Deterministic Tree Languages. *CSL*: 573–584 (2005).

16 F. Murlak. The Wadge Hierarchy of Deterministic Tree Languages. *Logical Methods in Comput. Sci.*, 4(4), Paper 15.

17 F. Murlak. Weak Index vs Borel Rank. *STACS*: 573–584 (2008).

18 D. Niwiński, I. Walukiewicz. A Gap Property of Deterministic Tree Languages. *Theor. Comput. Sci.* 303: 215–231 (2003).

19 D. Niwiński, I. Walukiewicz. Deciding Nondeterministic Hierarchy of Deterministic Tree Automata. *Electr. Notes Theor. Comput. Sci.* 123: 195–208 (2005).

20 M. O. Rabin. Decidability of second-order theories and automata on infinite trees. Trans. Amer. Soc. **141**: 1–35 (1969).

21 V. Selivanov. Wadge Degrees of ω-Languages of Deterministic Turing Machines. *Theoret. Informatics Appl.*, 37: 67–83 (2003).

22 J. Skurczyński. The Borel Hierarchy is Infinite in the Class of Regular Sets of Trees. *Theoret. Comput. Sci.* 112: 413–418 (1993).

23 W. W. Wadge. *Reducibility and Determinateness on the Baire Space.* Ph.D. Thesis, Berkeley (1984).

24 K. Wagner. On ω-Regular Sets. *Inform. and Control* 43: 123–177 (1979).