Relating hierarchies of word and tree automata

(extended abstract)

Damian Niwiński* Igor Walukiewicz*

Institute of Informatics, Warsaw University
Banacha 2, 02-097 Warsaw, POLAND
{niwinski,igw}@mimuw.edu.pl

Abstract

For an ω-word language L, a derived tree language $\mathrm{Path}(L)$ is the language of trees having all their paths in L. We consider the hierarchies of deterministic automata on words and nondeterministic automata on trees with Rabin conditions in chain form. We show that L is on some level of the hierarchy of deterministic word automata iff $\mathrm{Path}(L)$ is on the same level of the hierarchy of nondeterministic tree automata.

Subject classification: automata and formal languages, logic in computer science, verification

*Supported by Polish KBN grant No. 8 T11C 002 11

1 Introduction

For an ω-regular set of infinite words $L \subseteq \Sigma^\omega$, we consider the *derived language* of infinite binary trees, $\mathrm{Path}(L)$, consisting of the trees all of whose paths are in L. In this paper, we address the following question: what is the connection between the complexity of L and that of $\mathrm{Path}(L)$? Here, by the complexity of a language of ω-words or trees we understand the minimal complexity (in appropriate sense) of an automaton recognizing this language.

One motivation for considering this question is the following. In application of automata to program verification, a language L usually represents some property of computations of a program, e.g. expressed in a temporal logic of linear time. Now, we are often interested in a more general issue of whether all computations of a program satisfy the given property. Representing all possible computations of a program by a tree we are led to consider the derived tree language $\mathrm{Path}(L)$.

In most of the cases our language L would be an ω-regular language. It is easy to transform a deterministic automaton recognising L into a *deterministic* automaton recognising $\mathrm{Path}(L)$. However, we may a priori hope that using nondeterminism in some clever way we can simplify the automaton for $\mathrm{Path}(L)$.

In [6], Kupferman, Safra and Vardi investigated this question and showed a result that refutes such a hope in a special case of Büchi automata. More specifically, these authors show that a derived tree language is recognizable by a *nondeterministic* Büchi tree automaton if and only if the original language of ω–words is recognizable by a *deterministic* Büchi automaton. Read under the negation, the implication "only if" says that nondeterministic Büchi tree automata do not recognize more languages of the form $\mathrm{Path}(L)$ than they must. In yet another words, if an ω-regular language L cannot be recognized by a deterministic Büchi automaton, its inherent difficulty persists in $\mathrm{Path}(L)$.

In the present paper we show that a similar phenomenon occurs for *all* ω-regular languages. To make the statement precise, we need somehow to measure the complexity of languages L and $\mathrm{Path}(L)$. This can be done using hierarchies classifying ω-regular languages of trees or words. Our choice here (advocated below) is to consider the hierarchies induced by an index of a Mostowski acceptance condition (also called Rabin chain condition or parity condition[1]).

The Mostowski condition is given by a function $\Omega : Q \to \mathbb{N}$ assigning to each state of the automaton a natural number called its *priority*. Then a computation path is accepting if $\limsup_{n\to\infty}$ of all the priorities occurring along the path is even, in other words, the highest priority persisting

[1]To our knowledge, this condition was introduced by Wagner [15] and then rediscoverd by several authors. Mostowski [7] was the first to apply it to automata on trees. The actual phrasing, up to orientation, is due to Emerson and Jutla [3].

infinitely often is even. The index of a condition $\Omega : Q \to \mathbb{N}$ is the pair (ι, n) where ι and n are, respectively, the minimal and maximal values of the function Ω. By the nature of Mostowski condition, one can always scale Ω down so that ι is either 0 or 1. The concept of an index naturally gives rise to the hierarchies of ω-regular languages of words and trees recognized by automata equiped with Mostowski condition. Here we shall focuse on the hierarchies induced by *deterministic* automata on ω-words and *nondeterministic* automata on trees. Both hierarchies are known to be infinite and to exhaust the classes of ω-regular languages and regular sets of infinite trees, respectively. Moreover, these hierarchies refine the analgous hierarchies induced by the Rabin acceptance criterion. (Note that a hierarchy of *nondeterministic* automata on ω-words is uninteresting as it collapses at the level of Büchi automata, i.e. automata of the Mostowski index (1,2).)

Our main result is that Path(L) can be recognized by a nondeterministic tree automaton of Mostowski index (ι, n) iff L can be recognized by a deterministic automaton of the same index. That is, L and Path(L) are on the same level in the corresponding hierarchies.

Let us comment on the advantages of the Mostowski criterion compared to other acceptance criteria considered in the literature, and on the relevance of the aforementioned hierarchies.

The Mostowski condition is as powerful as Muller condition for nondeterministic automata on trees [7] and deterministic automata on ω-words [16]. Yet, Mostowski condition is much easier to work with. This is mainly because this is the unique condition that admits memoryless winning strategies for both of the players in infinite games with perfect information. This fact allows for a relatively easy proof of Rabin's Complementation Lemma ([8], cf. [14]); it is also used in many other applications of automata. The other reason for considering Mostowski condition is that when translating the μ-calculus into automata one naturally ends up with automata with Mostowski condition [3, 4]. Finally, as we have mentioned above, the hierarchies induced by the Mostowski condition are more subtle than those induced by the Rabin condition (c.f. Theorem 19); they also enjoy nice symmetry.

The aforementioned hierarchies are interesting for at least two reasons. They allow to classify the expressive power of formalisms. For example, one can better understand the expressive power of some logic of programs by translating it into automata of some level of the hierarchy. The second reason comes from considering complexity problems. For example, the general framework for solving satisfiability problems for logics of programs is to translate them into automata and then to do the emptiness check. Up to current knowledge, the deterministic complexity of the emptiness check depends exponentially on the size of the index (in general it is NP-complete for Rabin automata [2] and in NP$\cap$co-NP for Mostowski automata [1]).

Our work was directly inspired by the work of Kupferman, Safra and

Vardi [6], but our proof is different from theirs. For our proof we need a characterization of the index of an ω-word language in terms of properties of the graph of a deterministic automaton recognising the language. This characterization is related with the work of Wagner [16]. One may see our characterization as a simplification of his characterization for the restricted case of Mostowski conditions. Actually our characterization of Mostowski index can be transformed into a polynomial time algorithm computing the index of an ω–regular language (i.e. a minimal index of an automaton recognizing L). This is related with the work of Krishnan, Puri and Brayton [5] who show that computing the Rabin index of a language presented by a deterministic Rabin automaton is NP-complete. On the other hand, solving the same question is polynomial if the winning condition of the automaton is given in Muller form (c.f. Wilke and Yoo [17]).

The plan of the paper is as follows. We start with the preliminary section introducing automata and the hierarchies. (The relations between the hierarchies are described in the appendix.) In the next section, we show a simple characterization of the index of a ω-language in terms of properties of the graph of a deterministic Mostowski automaton recognizing the language. In the final section we use this characterization to show the main theorem.

2 Preliminaries

2.1 Words and trees

The set of natural numbers is denoted by ω or by $\mathbb{N}$. For a set X, X^* is the set of finite words over X, including the empty word ε, and X^ω is the set of mappings from ω to X usually referred to as *infinite words*. The *length* of a finite word w is denoted by $|w|$; note that $|\varepsilon| = 0$. We write $v \leq w$ to mean that v is an *initial segment* of w. For $u \in X^\omega$, we let $\mathrm{Inf}(u) = \{x : (\forall m \exists n > m)\, u(n) = x\}$ be the set of elements appearing infinitely often in u.

A nonempty subset T of X^* closed under initial segments is called a *tree*. The elements of T are called *nodes*, the $\leq$-maximal nodes are *leaves* and ε is the *root* of T. If $u \in T$, $x \in X$, and $ux \in T$ then ux is an *immediate successor* of u in T. An infinite sequence $P = (w_0, w_1, \ldots)$ such that $w_0 = \varepsilon$ and, for each m, w_{m+1} is an immediate successor of w_m is called a *path* in T.

If Σ is an arbitrary set and T is a tree then a mapping $t : T \to \Sigma$ is called an Σ-*valued* tree or shortly a Σ-*tree*; in this context T is the *domain* of t denoted by $T = \mathrm{dom}(t)$. We say "root of t", "path in t" etc., referring to the corresponding objects in $\mathrm{dom}(t)$. If $P = (w_0, w_1, \ldots)$ is a path in t, we denote by $t(P)$ the *labelling of* P, i.e., the ω–word $t(w_0)t(w_1)\ldots$ in Σ^ω. Note that $\mathrm{Inf}(t(P))$ is the set of values that occur infinitely often on the path P.

We now fix a finite alphabet Σ. For notational convenience, we shall focus in this paper on full binary trees over Σ, i.e., the Σ-trees with $\mathrm{dom}(t) = \{1,2\}^*$. Thus, every node $w \in \mathrm{dom}(t)$ has exactly two successors $w1$ and $w2$. Let T_Σ be the collection of all such trees.

The following is the key concept of our paper.

Definition 1 Let $L \subseteq \Sigma^\omega$. The *path tree language derived from* L is defined by

$$\mathrm{Path}(L) = \{t \in T_\Sigma : \text{ for each path } P \text{ in } t,\, t(P) \in L\}$$

We call a set $M \subseteq T_\Sigma$ a *path tree language* whenever $M = \mathrm{Path}(L)$, for some $L \subseteq \Sigma^\omega$.

Examples Let $\Sigma = \{a,b\}$, and let $L_1 = \{a,b\}^* a^\omega$, $L_2 = (\{a,b\}^* b)^\omega$ (for simplicity we abbreviate $\{s\}$ by s). Then $\mathrm{Path}(L_1)$ is the set of trees such that on each path b occurs only finitely many times, while $\mathrm{Path}(L_2)$ is the set of trees such that on each path b occurs infinitely often.

Not every set of trees is a path language, for example the set of trees t, such that, $t(w) = a$ for at least one w, is not.

2.2 Automata

Automata on words

A finite automaton on infinite words can be presented as a tuple $\mathcal{A} = \langle \Sigma, Q, q_0, Tr, Acc\rangle$, where Σ is a finite alphabet, Q is a finite set of *states*, q_0 is an *initial state*, $Tr \subseteq Q \times \Sigma \times Q$ is a set of *transitions*, and $Acc \subseteq Q^\omega$ is an acceptance criterion, usually given in some special finitary form.

A *run* of the automaton $\mathcal{A}$ on an infinite word $u \in \Sigma^\omega$ can be presented as an infinite word $r \in Q^\omega$ such that $r(0) = q_0$, and $\langle r(m), u(m), r(m+1)\rangle \in Tr$, for every $m < \omega$. A run is *accepting* if it belongs to Acc. A word $u \in \Sigma^\omega$ is *accepted* by $\mathcal{A}$, if there exists an accepting run of $\mathcal{A}$ on u. The set of all accepted words is denoted $L(\mathcal{A})$.

An automaton $\mathcal{A}$ as above is *deterministic* whenever Tr is a partial function from $Q \times \Sigma$ to Q, i.e., for every $q \in Q$, $\sigma \in \Sigma$, there is at most one q' such that $\langle q, \sigma, q'\rangle \in Tr$. Note that a deterministic automaton can have at most one run on a given word $u \in \Sigma^\omega$.

Several kinds of automata have been considered in the literature according to the actual form of the acceptance criterion. A *Büchi criterion* is given by a set $F \subseteq Q$, and the corresponding set Acc consists of those r for which $\mathrm{Inf}(r) \cap F \neq \emptyset$. A *Muller criterion* is given by a family $\mathcal{F} \subseteq \wp(Q)$, and $Acc = \{r : \mathrm{Inf}(r) \in \mathcal{F}\}$. A *Rabin criterion* is given by family $\{(L_1, U_1), \ldots, (L_n, U_n)\}$, $L_i, U_i \subseteq Q$, and then $Acc = \{r : (\exists i)\ \mathrm{Inf}(r) \cap L_i = \emptyset \text{ and } \mathrm{Inf}(r) \cap U_i \neq \emptyset\}$. Note that Büchi criterion can be presented as a special case of Rabin criterion (namely, $\{\emptyset, F)\}$), and the last as a special case of Muller criterion.

In this paper we focus on a yet another criterion that we call the *Mostowski acceptance criterion.* (This criterion has been often referred to as Rabin criterion in chain form.) Mostowski criterion can be presented by a *priority function* $\Omega : Q \to \mathbb{N}$, and Acc consists of those r for which $\limsup_{n\to\infty} \Omega(r(n))$ is *even*; in other words, the highest priority repeating infinitely often is even. It is not difficult to see that this can be represented by a Rabin criterion given by a family $(\{q : \Omega(q) \text{ is odd and } \geq i+1\}, \{q : \Omega(q) \text{ is even and } \geq i\})$, where i ranges over even numbers less than or equal to $\max(\Omega(Q))$.

According to the actual form of the acceptance criterion we shall refer to Büchi automata, Mostowski automata, etc.

Two automata $\mathcal{A}_1, \mathcal{A}_2$ are *semantically equivalent* whenever $L(\mathcal{A}_1) = L(\mathcal{A}_2)$. By the remarks above, for every Büchi, Rabin or Mostowski automaton, there is an equivalent Muller automaton. Moreover, the following facts are well–known (see e.g. [13]).

Theorem 2
Every Muller automaton can be transformed into an equivalent nondeterministic Büchi *automaton, and into an equivalent* deterministic Mostowski *automaton, but in general not into a* deterministic Büchi *automaton.*

Automata on trees

An automaton over binary trees can be presented by $\mathcal{A} = \langle \Sigma, Q, q_0, Tr, Acc\rangle$, where all the items are as in automaton on words except that $Tr \subseteq Q \times \Sigma \times Q \times Q$. A *run* of $\mathcal{A}$ on a tree $t \in T_\Sigma$ is a Q–valued tree $r : \{1,2\}^* \to Q$ such that $r(\varepsilon) = q_0$, and, for each $w \in \mathrm{dom}(r)$, the tuple $\langle r(w), t(w), r(w1), r(w2)\rangle$ is in Tr.

A path P in a run r is *accepting* if $r(P) \in Acc$; a run is accepting if so are all its paths. The set of trees *accepted* by $\mathcal{A}$, denoted $L(\mathcal{A})$, consists of those $t \in T_\Sigma$ for which there exists an accepting run of $\mathcal{A}$ on t.

Two automata $\mathcal{A}_1, \mathcal{A}_2$ are *semantically equivalent* whenever $L(\mathcal{A}_1) = L(\mathcal{A}_2)$.

Similarly as for automata on infinite words, we can consider different kinds of tree automata according to the form of the acceptance criterion. Since the concept of acceptance criterion is the same as for automata on words, we have again some trivial inclusions. Moreover, Mostowski [7] established the following.

Theorem 3
Any Muller automaton can be transformed into an equivalent Mostowski automaton.

This is the furthest that the analogy with Theorem 2 can go. In contrast to automata on words, tree automata cannot in general be determinized and the expressive power of nondeterministic Büchi tree automata is weaker than that of Muller automata, and incomparable with that of deterministic Muller tree automata [12].

2.3 Index hierarchies

It is an acceptance criterion that gives rise to the expressive power of automata; the criteria also induce several semantical hierarchies. Here we will just restrict to the hierarchy of Mostowski conditions. In the appendix we summarise the relationship of this hierarchy with the hierarchy of Rabin conditions and fixpoint alternations.

Definition 4 An *index* of a Mostowski criterion given by the function Ω is a pair (ι, n), where ι and n are the minimal and the maximal values of Ω, respectively.

We say that a language $L \subseteq \Sigma^\omega$ is ι-n-*feasible* iff L can be recognized by a *deterministic* Mostowski automaton with the condition of index (ι, n). Otherwise, L is ι-n*–unfeasible*.

We say that a set of trees $M \subseteq \mathcal{T}_\Sigma$ is ι-n-*feasible* iff it can be recognized by a nondeterministic Mostowski automaton on trees with the condition of index (ι, n), and is ι-n-*unfeasible* otherwise.

Note that we express feasibility of sets of words in terms of deterministic automata, and feasibility of sets of trees in terms of general, i.e., nondeterministic automata.

It turns out that the hierarchy of indices induces truly infinite hierarchy of languages. Let, for $n < \omega$, $\Sigma_n = \{0, 1, \ldots, n\}$. Consider the following families of languages, $M_n, N_n \subseteq \Sigma_n^\omega$:

$$\begin{aligned} M_n &= \{u \in \Sigma_n^\omega : \limsup_{n\to\infty} u(n) \text{ is } \textit{even} \} \\ N_n &= \{u \in \Sigma_n^\omega : \limsup_{n\to\infty} u(n) \text{ is } \textit{odd} \} \end{aligned}$$

The following results have been originally stated in different terms but the actual phrasing follows easily from the correspondence between Rabin and Mostowski indices (see the Appendix).

Theorem 5 (Wagner [15])
For every $n < \omega$: (i) M_n is 0-n-feasible but 1-$(n+1)$-unfeasible; (ii) N_n is 1-$(n+1)$-feasible but 0-n-unfeasible.

Theorem 6 (Niwiński [9, 11])
For every $n < \omega$: (i) Path(M_n) is 0-n-feasible but 1-$(n+1)$-unfeasible; (ii) Path(N_n) is 1-$(n+1)$-feasible but 0-n-unfeasible.

We end this section by an observation that is an easy part of our main result.

Proposition 7 For a *deterministic* Mostowski automaton $\mathcal{A}$, $\mathrm{Path}(L(\mathcal{A}))$ is recognized by a (deterministic) Mostowski tree automaton of the same index.

Proof
If $\mathcal{A} = \langle \Sigma, Q, q_0, Tr, Acc\rangle$ is a deterministic automaton recognizing L, then the automaton for $\mathrm{Path}(L)$ is $\mathcal{A}' = \langle \Sigma, Q, q_0, Tr', Acc\rangle$, with all the same components as $\mathcal{A}$ but $Tr' = \{\langle q, \sigma, q', q'\rangle : \langle q, \sigma, q'\rangle \in Tr\}$. □

3 Flower Lemma

In this section we will show a connection between the Mostowski index of an ω-word language and the shape of a deterministic Mostowski automaton recognizing the language. Roughly speaking, we will show that in the graph of an automaton recognizing a "hard" language there must be a subgraph, called a flower, "responsible" for this hardness.

Definition 8 Let $\mathcal{A} = \langle \Sigma, Q, q_0, Tr, \Omega\rangle$ be a Mostowski automaton on words. The *graph of* $\mathcal{A}$ is the graph obtained by taking Q as the set of vertices and adding an edge from q to q' whenever $\langle q, a, q'\rangle \in Tr$, for some letter a.

A *path* in a graph is a sequence of vertices $v_1, \ldots, v_j$, such that, for every $i = 1, \ldots, j-1$ there is an edge from v_i to v_{i+1} in the graph. A *maximal strongly connected component* of a graph is a maximal subset of vertices of the graph, such that, for every two vertices v_1, v_2 in the subset, there is a path from v_1 to v_2 and from v_2 to v_1.

For an integer k, a *k-loop* in $\mathcal{A}$ is a path $v_1, \ldots, v_j$ in the graph of $\mathcal{A}$, such that, $v_1 = v_j$, $j > 1$ and $k = \max\{\Omega(v_i) : i = 1, \ldots, j\}$. Please observe that a k-loop must necessarily go through at least one edge.

Given integers m and n, a state $q \in Q$ is a *m-n-flower* in $\mathcal{A}$ if for every $k \in \{m, \ldots, n\}$ there is, in the graph of $\mathcal{A}$, a k-loop containing q.

First, we need an operation allowing us to scale unnecessary indices up.

Definition 9 For a Mostowski automaton $\mathcal{A}$ as above and an integer i we define the automaton $\mathcal{A}\uparrow^i$ obtained from $\mathcal{A}$ by *lifting the index i*. The automaton $\mathcal{A}\uparrow^i$ has all the same components as $\mathcal{A}$ except for the priority function $\Omega\uparrow^i$ defined in the following steps:

1. Take the set $Q_{\leq i} = \{q \in Q : \Omega(q) \leq i\}$. Let $G_{\leq i}$ denote the graph of $\mathcal{A}$ restricted to the states from $Q_{\leq i}$.

2. If S is a trivial maximal strongly connected component of the graph $G_{\leq i}$, i.e., a component consisting of one state without a self loop, then let $\Omega\uparrow^i(q) = i + 1$ for q being the unique state in S.

3. If S is a nontrivial maximal strongly connected component of $G_{\leq i}$ and S contains only states of priorities strictly smaller than i then let $\Omega{\uparrow}^i\,(q) = \Omega(q) + 2$ for every $q \in S$.

4. If a state q is in none of the above components then let $\Omega{\uparrow}^i\,(q) = \Omega(q)$.

Lemma 10 For every deterministic Mostowski automaton on words $\mathcal{A}$ and every integer i: $L(\mathcal{A}) = L(\mathcal{A}{\uparrow}^i)$

Proof
Let $\mathcal{A} = \langle \Sigma, Q, q_0, Tr, \Omega : Q \to \{0, \ldots, n\}\rangle$ be a deterministic Mostowski automaton on words. Consider an infinite word $w \in \Sigma^\omega$. There is at most one run of $\mathcal{A}$ on w and if it exists it is also a run of $\mathcal{A}{\uparrow}^i$. This run determines the set of states Q_w that are met infinitely often on the run. Let $k = \max\{\Omega(q) : q \in Q_w\}$.

Suppose $k > i$. Every state of priority k in Q_w has still the priority k in $\mathcal{A}{\uparrow}^i$. Observe that if $\Omega{\uparrow}^i\,(q) \neq \Omega(q)$ then $\Omega{\uparrow}^i\,(q) \leq i + 1$. Hence $k = \max\{\Omega{\uparrow}^i\,(q) : q \in Q_w\}$ as well. So, in this case, we have that $\mathcal{A}$ accepts w iff $\mathcal{A}{\uparrow}^i$ accepts w.

If $k \leq i$ then all the states in Q_w belong to the same strongly connected component of the graph $G_{\leq i}$. According to the definition of $\mathcal{A}{\uparrow}^i$ (see Definition 9), all of the priorities of the states in Q_w are either increased by 2 or left unchanged. In both cases, $\mathcal{A}$ accepts w iff $\mathcal{A}{\uparrow}^i$ does. $\square$

Lemma 11 Let $\mathcal{A}$ be a deterministic Mostowski automaton and $i \in \mathbb{N}$. Let $\mathcal{B} = \mathcal{A}{\uparrow}^0{\uparrow}^1 \ldots {\uparrow}^i$, i.e., $\mathcal{B}$ is an automaton obtained from $\mathcal{A}$ by lifting indices from 0 to i. If a state q has the priority m in $\mathcal{B}$ then q is a m-i-flower in $\mathcal{B}$.

Proof
We prove the claim by induction on i. Let us denote by $\Omega_{\leq i}$ the priority function of the automaton $\mathcal{A}{\uparrow}^0{\uparrow}^1 \ldots {\uparrow}^i$.

Consider first $i = 0$. If $\Omega_{\leq 0}(q) > 0$ then the claim is trivial, so suppose $\Omega_{\leq 0}(q) = 0$.

Let, as in Definition 9, $G_{\leq 0}$ be the graph of $\mathcal{A}$ restricted to the states that have the priority 0. There must be a 0-loop in $G_{\leq 0}$ containing q, otherwise the priority of q would be increased. Call this loop P. Clearly, the operation ${\uparrow}^0$ increases the priority of no node on this loop, and thus P remains a 0-loop in the automaton $\mathcal{A}{\uparrow}^0$. Therefore, q is a 0-0-flower.

Now, let $i > 0$ and suppose the result holds for $i - 1$. Let $\Omega_{\leq i}(q) = m$. Again, if $m > i$, the claim is trivial, so we can assume $m \leq i$.

Let $G'_{\leq i}$ be the graph of $\mathcal{A}$ restricted to the states that have the priority not greater than i in the automaton $\mathcal{A}{\uparrow}^0{\uparrow}^1 \ldots {\uparrow}^{i-1}$. Let $m' = \Omega_{\leq i-1}(q)$. By the induction hypothesis, q is an m'-$(i-1)$-flower in $\mathcal{A}{\uparrow}^0{\uparrow}^1 \ldots {\uparrow}^{i-1}$.

We consider two cases.

1. Suppose $m' = m$, i.e., the operation $\uparrow^i$ has not changed the priority of q. This is possible only if there existed in $G'_{\leq i}$ an i-loop containing q. Call this loop P. Clearly, the operation $\uparrow^i$ has not shifted the priority of any other node in the maximal strongly connected component of $G'_{\leq i}$ containing q. In particular, P continues to be an i-loop in $\mathcal{A}\uparrow^0\uparrow^1 \ldots \uparrow^i$, and every j-loop in $\mathcal{A}\uparrow^0\uparrow^1 \ldots \uparrow^{i-1}$ containing q, for $j < i$, remains a j-loop in $\mathcal{A}\uparrow^0\uparrow^1 \ldots \uparrow^i$. Since q is an m-$(i-1)$-flower in $\mathcal{A}\uparrow^0\uparrow^1 \ldots \uparrow^{i-1}$, it is an m-i-flower in $\mathcal{A}\uparrow^0\uparrow^1 \ldots \uparrow^i$.

2. $m' < m$. The only possibility is $m = m' + 2$. By the definition of lifting (Definition 9), this means that the operation $\uparrow^i$ has shifted by 2 the priorities of all the states in the maximal strongly connected component of $G'_{\leq i}$ containing q. Thus, every loop containing q that was a j-loop in $\mathcal{A}\uparrow^0\uparrow^1 \ldots \uparrow^{i-1}$, with $j < i$, has become a $(j+2)$-loop in $\mathcal{A}\uparrow^0\uparrow^1 \ldots \uparrow^i$. Since q is an m'-$(i-1)$-flower in $\mathcal{A}\uparrow^0\uparrow^1 \ldots \uparrow^{i-1}$, it is an m-$(i+1)$-flower in $\mathcal{A}\uparrow^0\uparrow^1 \ldots \uparrow^i$. Hence *a fortiriori* an m-i-flower.

□

Definition 12 We say that a language $L \subseteq \Sigma^\omega$ *admits* an m-n-flower if there exists a deterministic Mostowski automaton $\mathcal{A}$, such that, $L = L(\mathcal{A})$ and $\mathcal{A}$ has an m-n-flower q for some q not a useless state in $\mathcal{A}$ (i.e. q occurring in some accepting run of $\mathcal{A}$).

Lemma 13 (Flower Lemma) For every $n \in \mathbb{N}$ and $L \subseteq \Sigma^\omega$: (1) if L is 1-$(n+1)$-unfeasible then L admits a $2i$-$(2i+n)$-flower, for some i; (2) if L is 0-n-unfeasible then L admits a $(2i+1)$-$(2i+1+n)$-flower, for some i.

Proof
We prove 1 and leave the case 2 to the reader.

Let $L = L(\mathcal{A})$ for some deterministic Mostowski automaton $\mathcal{A}$. We can assume without a loss of generality that $\mathcal{A}$ has no useless states. Clearly, for arbitrary k, the operation $\uparrow^k$ preserves this property.

Let M be the maximal value of the priority function of $\mathcal{A}$. Choose i such that $2i + n > M$. Consider the automaton $\mathcal{B} = \mathcal{A}\uparrow^0\uparrow^1 \ldots \uparrow^{2i+n}$. By Lemma 10, $L(\mathcal{B}) = L$. If we can find a state q with the priority in $\mathcal{B}$ less or equal to $2i$ then we are done since, by Lemma 11, q is a $2i$-$(2i+n)$-flower in $\mathcal{B}$, and by the remark above q is not useless.

Suppose to the contrary that the minimal value of the priorities of all states in $\mathcal{B}$ is $j > 2i$. It should be clear that the maximal priority of a state in $\mathcal{B}$ cannot exceed $2i + n + 1$. (The operation $\uparrow^k$ can change the priority of a state to at most $k + 1$.) By adding, if necessary, some dummy states we obtain that L is $(2i+1)$-$(2i+1+n)$-feasible. By scaling the priorities down we conclude that L is 1-$(n+1)$-feasible, contradicting the assumption. □

For a given automaton $\mathcal{A}$ and integer i one can calculate $\mathcal{A}\uparrow^i$ in the time proportional to the number of transitions in $\mathcal{A}$. Hence one can compute $\mathcal{A}\uparrow^1 \dots \uparrow^n$ in time $\mathcal{O}(n|\mathcal{A}|)$. As we can easily limit the range of the Ω function to be at most twice as big as the number of states in $\mathcal{A}$ we get:

Corollary 14 The problem of establishing the index of the language accepted by an automaton with a Mostowski condition can be solved in time $\mathcal{O}(|\mathcal{A}|^2)$.

4 The main result

Lemma 15

1. If a language $L \subseteq \Sigma^\omega$ admits a $2m$-$(2m+n)$-flower for some m, then Path(L) is 1-$(n+1)$-unfeasible.
2. If a language $L \subseteq \Sigma^\omega$ admits a $(2m+1)$-$(2m+1+n)$-flower for some m, then Path(L) is 0-n-unfeasible.

Proof
We shall prove 1, the case of 2 is analogous.

Let $\mathcal{A}$ be a deterministic Mostowski automaton recognizing L and let q be a $2m$-$(2m+n)$-flower in $\mathcal{A}$, where moreover q is not useless.

Suppose to the contrary that Path(L) is 1-$(n+1)$-feasible, and let:

$$\mathcal{C} = \langle \Sigma, Q^c, q_0^c, Tr^c, \Omega^c : Q^c \to \{1, \dots, n+1\} \rangle$$

be a nondeterministic Mostowski automaton of index $(1, n+1)$ recognizing Path(L). Without a loss of generality we may assume that there are no useless states in $\mathcal{C}$.

¿From $\mathcal{C}$ we shall construct a tree automaton $\mathcal{D}$ of the same index but over the alphabet $\{0, \dots, n\}$ that will recognize Path(M_n). This however will contradict Theorem 6 proving that an automaton $\mathcal{C}$ can not exist.

To this end, we shall use our flower q in $\mathcal{A}$. For every $i = 0, \dots, n$ fix a word w_i, s.t., there is a $(2m+i)$-loop in $\mathcal{A}$ starting and ending in q, and the labelling of this loop is w_i. Clearly, this is possible by the definition of a flower.

Apart from $w_0, \dots, w_n$, fix also a finite word v that is a labelling of a path in $\mathcal{A}$ from the initial state to q. For notational convenience we assume that the length of v as well as all w_i $(i = 1, \dots, n)$ is k.

Before defining formally an automaton for Path(M_n), we explain its idea. It is based on a transformation of trees over the alphabet $\{0, \dots, n\}$ into trees over the alphabet Σ. This transformation essentially replaces a node labelled by $i \in \{0, \dots, n\}$ by a finite path labelled by w_i. Then, when examining a tree t (over the alphabet $\{0, \dots, n\}$), our automaton $\mathcal{D}$ will mimic the work of the automaton $\mathcal{C}$ on the transformed tree. This will be done in such a way that a computation of $\mathcal{D}$ along a path labelled by $i_0 i_1 i_2 \dots$

($i_\ell \in \{0, \dots, n\}$) will essentially simulate a computation of $\mathcal{C}$ along a path labelled $vw_{i_0}w_{i_1}w_{i_2}\dots$

Let a function $h : \{1,2\}^* \to \{1,2\}^*$ be defined by:

$$h(d_1 \dots d_i) = 1^{k-1}d_1 1^{k-1} \dots d_i 1^{k-1}$$

Let us denote by R the smallest tree containing the range of h.

As the next step we define an operation transforming a labelled tree $\tau : \{1,2\}^* \to \{0, \dots, n\}$ into a labelled tree $\tau^\sharp : R \to \Sigma$. For arbitrary $v \in \{1,2\}^*$, $d \in \{1,2\}$ and $i \in \{0, \dots, k-1\}$ we let:

$$\begin{aligned} \tau^\sharp(1^i) &= \text{the } (i+1)\text{th letter of } v \\ \tau^\sharp(h(v)d1^i) &= \text{the } (i+1)\text{th letter of } w_{\tau(v)} \end{aligned}$$

The key property of this transformation is the following:

> for every path $P = d_1d_2\dots$ in $\mathrm{dom}(\tau)$ ($d_\ell \in \{1,2\}$), there is a path $P^\sharp = 1^{k-1}d_1 1^{k-1}d_2 1^{k-1}\dots$ in $\mathrm{dom}(\tau^\sharp)$ with the labelling $\tau^\sharp(P^\sharp) = vw_{\tau(\varepsilon)}w_{\tau(d_1)}w_{\tau(d_1d_2)}\cdots$

Now, it follows form the choice of the words v, $w_0, \dots, w_n$, that an infinite word of the form $vw_{i_1}w_{i_2}\dots$ is in $L = L(\mathcal{A})$ iff $\limsup_{\ell\to\infty} i_\ell$ is even. In particular, we note the following.

Observation 15.1 $\tau^\sharp(P^\sharp) \in L$ iff $\tau(P) \in M_n$.

Now, we define the automaton:

$$\mathcal{D} = \langle \Sigma^d, Q^d = Q^c \times \{1, \dots, n+1\} \cup \{q_s\}, q_s, Tr^d, \Omega : Q^d \to \{1, \dots, n+1\}\rangle$$

where $\Sigma^d = \{0, \dots, n\}$ and the meanings of the other components are defined below.

The state q_s is the initial state and it will be used only in the starting move of $\mathcal{D}$. For the function Ω^d, we let $\Omega^d(q,i) = i$. We let $\Omega^d(q_s) = 1$ but this value does not really matter.

Before defining the transition function let us introduce an auxiliary notion. We say that a pair of states $q_l, q_r \in Q^c$ is *reachable* from a state $q \in Q^c$ on a word $w = a_1, \dots, a_k$ iff there is a sequence $q = q_1, \dots, q_{k+1} = q_l$ and a sequence $q'_2, \dots, q'_{k+1} = q_r$ with $(q_{j+1}, q'_{j+1}) \in Tr^c(q_j, a_j)$. We say that (q_l, q_r) is *i-reachable* from q on w if $i = \max(\Omega(q_1), \dots, \Omega(q_k))$.

Now, $((q_l, i), (q_r, i)) \in Tr^d(q_s, a)$ if (q_l, q_r) is i-reachable from the initial state q_0^c of $\mathcal{C}$ on the word v. Also, $((q_l, j), (q_r, j)) \in Tr^d((q,i), m)$ if (q_l, q_r) is j-reachable from q on the word w_m.

To prove the claim it is enough to show that $L(\mathcal{D}) = \mathrm{Path}(M_n)$. With the help of the operation $\sharp$ introduced above, we can state the correspondence between $\mathcal{C}$ and $\mathcal{D}$:

Observation 15.2 For every tree $\tau : \{1,2\}^* \to \{0,\ldots,n\}$: $\tau \in L(\mathcal{D})$ iff $\tau^\sharp$ can be extended to a full binary tree that is accepted by $\mathcal{C}$.

Recall that $\mathcal{C}$ accepts a tree iff the labelling of every path of this tree is accepted by $\mathcal{A}$. So, if $\mathcal{C}$ accepts an extension of $\tau^\sharp$ then in particular $\tau^\sharp(P^\sharp) \in L(\mathcal{A})$, for each path P in $\mathrm{dom}(\tau)$. By 15.1, we conclude that $\tau \in \mathrm{Path}(M_n)$. We have proved $L(\mathcal{D}) \subseteq \mathrm{Path}(M_n)$.

For the converse inclusion, let $\tau \in \mathrm{Path}(M_n)$. Then clearly $\tau^\sharp$ can be extended to a tree in $\mathrm{Path}(L)$. The conclusion now follows from the hypothesis that $L(\mathcal{C}) = \mathrm{Path}(L)$. $\square$

Theorem 16
For every $L \subseteq \Sigma^\omega$ and $m, n \in \mathbb{N}$: L is m-n-feasible if and only if Path(L) is m-n-feasible.

Proof
The implication "only if" follows from Proposition 7.

Suppose L is m-n-unfeasible. The case of $n < m$ is trivial, so we can assume $m \leq n$.

Suppose m is even. Then clearly L is also 0-$(n-m)$-unfeasible since otherwise we could scale up the priorities in the hypothetical automaton by m. By the Flower Lemma, L admits a $(2i+1)$-$(2i+1+n-m)$-flower, for some i, and then by Lemma 15, $\mathrm{Path}(L)$ is 0-$(n-m)$-unfeasible. Then clearly it is also m-n-unfeasible, since otherwise we could scale down the hypothetical automaton by m.

If m is odd, we present it as $m = m' + 1$, and the remaining of the argument is similar. $\square$

It follows from our considerations on the connections between Mostowski and Rabin automata (see the appendix) that a similar statement can be made for the latter.

Corollary 17 For every $L \subseteq \Sigma^\omega$ and $n \in \mathbb{N}$, L can be recognized by a deterministic Rabin automaton of index n if and only if $\mathrm{Path}(L)$ can be recognized by a nondeterministic Rabin tree automaton of index n.

Finally let us remark on the problem of constructing a deterministic automaton recognising L from a nondeterministic tree automaton recognising $\mathrm{Path}(L)$. This can be done by constructing a nondeterministic automaton recognising L, determinising it and then applying our index lifting procedure from the flower lemma. The so–obtained deterministic automaton would have the least possible index.

References

[1] E. A. Emerson, C. Jutla, and A. Sistla. On model-checking for fragments of μ-calculus. In *CAV'93*, volume 697 of *LNCS*, pages 385–396, 1993.

[2] E. A. Emerson and C. S. Jutla. The complexity of tree automata and logics of programs. In *29th FOCS*, 1988.

[3] E. A. Emerson and C. S. Jutla. Tree automata, mu-calculus and determinacy. In *Proc. FOCS 91*, 1991.

[4] D. Janin and I. Walukiewicz. Automata for the μ-calculus and related results. In *MFCS '95*, volume 969 of *LNCS*, pages 552–562, 1995.

[5] S. Krishnan, A. Puri, and R. Brayton. Srtuctural complexity of ω-automta. In *STACS'95*, volume 900 of *LNCS*, 1995.

[6] O. Kupferman, S. Safra, and M. Vardi. Relating word and tree automata. In *11th IEEE Symp. on Logic in Comput. Sci.*, pages 322–332, 1996.

[7] A. W. Mostowski. Regular expressions for infinite trees and a standard form of automata. In A. Skowron, editor, *Fifth Symposium on Computation Theory*, volume 208 of *LNCS*, pages 157–168, 1984.

[8] A.W. Mostowski. Games with forbidden positions. Technical Report 78, University of Gdańsk, 1991.

[9] D. Niwiński. On fixed-point clones. In *Proc. 13th ICALP*, volume 226 of *LNCS*, pages 464–473, 1986.

[10] D. Niwiński. Fixed points vs. infinite generation. In *LICS '88*, pages 402–409, 1988.

[11] D. Niwiński. Fixed point characterization of infinite behaviour of finite state systems. *TCS*, 1997. to appear.

[12] M. Rabin. Weakly definable relations and special automata. In Y.Bar-Hillel, editor, *Mathematical Logic in Foundations of Set Theory*, pages 1–23. 1970.

[13] W. Thomas. Automata on infinite objects. In J. van Leeuven, editor, *Handbook of Theoretical Computer Science Vol.B*, pages 133–192. Elsevier, 1990.

[14] W. Thomas. Languages, automata, and logic. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 3. Springer-Verlag, 1997.

[15] K. Wagner. Eine topologische charakterisierung einiger klassen regulärer folgenmengen. *J. Inf. Process. Cybern. EIK*, 13:473–487, 1977.

[16] K. Wagner. On ω–regular sets. *Information and Control*, 43:123–177, 1979.

[17] T. Wilke and H. Yoo. Computing the Rabin index of a regular language of infinite words. *Information and Computation*, 130(1):61–70, 1996.

A Index hierarchies

In this section we summarise the relationship of the hierarchy of Mostowski conditions with the hierarchies of Rabin conditions and of fixpoint alternations.

The following is a refinement of a standard concept of an index of a Rabin automaton.

Definition 18 An *index* of a Rabin automaton (on trees or words) with a criterion $\{(L_1, U_1), \ldots, (L_n, U_n)\}$ is the number n. Moreover, we say that this index is *weakened by* $\emptyset$, if for some i, $L_i = \emptyset$. We also say that the index is *weakened "by all"*, if, for some i, $L_i \cup U_i$ equals to the set of all the states.

Intuitively, a pair $(\emptyset, U)$ introduces only positive constraint (no state is forbidden), and a pair (L, U) such that $L \cup U$ amounts to the set of all states Q introduces only negative constraint (semantically it is equivalent to the pair (L, Q)). Note that an automaton can have index n weakened both by $\emptyset$ and "by all" if the acceptance condition contains either two pairs corresponding to the both restrictions or a pair $(\emptyset, Q)$. An automaton with the empty acceptance condition has the index 0 which, by definition, may not be weakened by anything.

It should be clear that an automaton of index (ι, n) can be transformed into an equivalent automaton of index (ι', n') with any $\iota' \leq \iota$, $n \leq n'$, e.g. by adding some dummy states. Also, we can always "scale up" or "scale down" an automaton by shifting the priorities of all the states by the same even value. In particular, any automaton is equivalent to one with $\iota = 0$ or $\iota = 1$.

¿From the remark above on Mostowski vs. Rabin condition, it can be easily seen that a Mostowski automaton of index (ι, n) is equivalent, by simple rephrasing of the acceptance condition, to a Rabin automaton of index $\lfloor \frac{n}{2} \rfloor + \iota$. Moreover, if n is even, this index is weakened by $\emptyset$ (because then $\{q : \Omega(q) \text{ is odd and } \geq n+1\} = \emptyset$), and if $\iota = 0$, it is weakened "by all" (because the pair $(\{q : \Omega(q) \text{ is odd and } \geq 1\}, \{q : \Omega(q) \text{ is even and } \geq 0\})$, is equivalent to $(\{q : \Omega(q) \text{ is odd and } \geq 1\}, Q))$.

A converse transformation is also possible, but it generally requires an extension of a transition table and may cause an exponential blow up of the number of states of an automaton. We note the result only for tree automata. The similar can be shown for deterministic word automata as well.

Theorem 19
Any tree Rabin automaton $\mathcal{A}$ of index $n < \omega$ can be transformed into a Mostowski automaton $\mathcal{A}'$ of index $(1, 2n+1)$. Moreover,

1. *if the index of $\mathcal{A}$ is weakened "by all" then $\mathcal{A}'$ can be chosen with an index $(0, 2n-1)$;*
2. *if the index of $\mathcal{A}$ is weakened by $\emptyset$ then $\mathcal{A}'$ can be chosen with an index $(1, 2n)$;*
3. *if the index of $\mathcal{A}$ is weakened both "by all" and by $\emptyset$ then $\mathcal{A}'$ can be chosen with an index $(0, 2n-2)$.*

Proof
It has been shown in [11] that every Rabin tree automaton can be transformed into an equivalent fixed point term, and then back to a Rabin automaton ([11], Theorems 3.2 and 3.3) of the same index and with an acceptance condition in chain form (i.e. with $L_1 \supseteq L_2 \supseteq \ldots \supseteq L_n$). It is also shown there (Section 1.3) that the last can be (very easily) transformed into Mostowski automaton. The claims about indices follow directly from these constructions.

For a more direct argument see [14]. $\square$

As we can see, the hierarchy induced by Mostowski indices refines that of Rabin indices. This is one of the reasons why we have chosen here to work with Mostowski automata.

It turns out that the hierarchy of indices induces truly infinite semantical hierarchies both for deterministic automata on words, and nondeterministic automata on trees. Examples that witness the infinity of both hierarchies are based on the same idea which is inherent in the very concept of the Mostowski condition. Let, for $n < \omega$, $\Sigma_n = \{0, 1, \ldots, n\}$. Consider the following families of languages, $M_n, N_n \subseteq \Sigma_n^\omega$.

$$\begin{aligned} M_n &= \{u \in \Sigma_n^\omega : \limsup_{n\to\infty} u(n) \text{ is } \textit{even}\ \} \\ N_n &= \{u \in \Sigma_n^\omega : \limsup_{n\to\infty} u(n) \text{ is } \textit{odd}\ \} \end{aligned}$$

The following results have been shown by K. Wagner [15] and D. Niwiński [9, 11], respectively.

Theorem 20 ([15])
For every $n < \omega$: (i) M_n is 0-n-feasible but 1-$(n+1)$-unfeasible; (ii) N_n is 1-$(n+1)$-feasible but 0-n-unfeasible.

Theorem 21 ([9, 11])
For every $n < \omega$: (i) Path(M_n) is 0-n-feasible but 1-$(n+1)$-unfeasible; (ii) Path(N_n) is 1-$(n+1)$-feasible but 0-n-unfeasible.

The result for trees has been shown in [11] for Rabin automata, but the actual statement of Theorem 21 (Theorem 6 in Section 2.3) follows from Theorem 19. The original statement of the result of Wagner [15] was also slightly different, but we may note that this result has been actually reproved in our paper: Theorem 20 follows from Theorem 21 and Theorem 16.

Finally let us briefly describe the connections with the fixpoint calculus. Recall that fixed point terms (for binary trees over an alphabet Σ) are defined inductively as follows: variables $x_0, x_1, \ldots$ are fixed point terms, and if τ, τ' are fixed point terms, so are $\tau \cup \tau'$, $\sigma(\tau, \tau')$, for $\sigma \in \Sigma$, $\mu x_i.\tau$, and $\nu x_i.\tau$. Here variables are intended to range over sets of trees, $\cup$ is interpreted as union, σ as the constructor $\sigma(M_1, M_2) = \{t : t(\varepsilon) = \sigma,\ t_1 \in M_1 \text{ and } t_2 \in M_2\}$, and $\mu x_i.\tau$, $\nu x_i.\tau$, are interpreted as the least (respectively, the greatest) solution of the equation $x_i = \tau_i$. (For more information, see [11] or [9, 10].)

Fixed point terms can be structured into classes Σ^μ_n, Π^μ_n, defined inductively as follows. $\Sigma^\mu_0 = \Pi^\mu_0$ is the class of terms without μ and ν; Σ^μ_{n+1} is the closure of Π^μ_n under composition and μ, and Π^μ_{n+1} is the closure of Σ^μ_n under composition and ν. In [11] a correspondence is proved between these classes and classes of automata with refined Rabin indices. Using the correspondences of Theorem 19, the result of [11] can be rephrased as follows. For odd n, fixed point terms of the class Σ^μ_n are semantically equivalent (i.e. define the same set of trees) to the Mostowski automata of index $(1, n)$, and fixed point terms of the class Π^μ_n are equivalent to the automata of index $(0, n-1)$. For even $n > 0$, similar equivalences hold for the classes Σ^μ_n vs. index $(0, n-1)$ and Π^μ_n vs. index $(1, n)$.