A Playful Glance at Hierarchical Questions for Two-Way Alternating Automata

Jacques Duparc[1] and Alessandro Facchini[1,2]*

[1] University of Lausanne, Faculty of Business and Economics - ISI, University of Lausanne, CH-1015 Lausanne
[2] LaBRI, University of Bordeaux 1, 351 cours de la Libération, FR-33405 Talence cedex
{jacques.duparc, alessandro.facchini}@unil.ch

Abstract. Two-way alternating automata were introduced by Vardi in order to study the satisfiability problem for the modal μ-calculus extended with backwards modalities. In this paper, we present a very simple proof by way of Wadge games of the strictness of the hierarchy of Motowski indices of two-way alternating automata over trees.

Keywords : μ-calculus, backward modalities, Wadge games, topological complexity, parity games, two-way alternating tree automata, descriptive set theory.

1 Introduction

Since the seminal work of Büchi, the idea of translating a logic into appropriate models of finite-state automata on infinite words or infinite trees has become a central paradigm in the theory of verification of concurrent systems. One of the reasons is that this translation reduces the model-checking issue to the non emptyness problem for automata. Two hiearchies are classically used to measure the complexity of recognizable sets of infinite words or trees: the Mostowski index hierarchy, and the Wadge hierarchy. Since it reflects the depth of nesting of positive and negative conditions, the first hierarchy determines the combinatorial complexity of the recognizing automaton and therefore is closely related to the fixpoint alternation hierarchy of the modal μ-calculus, unveiling subtle connections between this logic, (parity) automata and games. The second one on the other hand captures the topological complexity of the languages accepted by such machines. Indeed from a topological point of view, an infinite tree is very similar to an infinite word, which is also very close to a real number. Moreover, evidences indicate that the two hierarchies are closely related. Thus, as is well said in [NW05], "understanding the structure of these hierarchies helps us to understand the trade-off between expressivness and efficiency in the model-checking method".

* Research supported by a grant from the Swiss National Science Foundation, n. 100011-116508: Project "Topological Complexity, Games, Logic and Automata".

For ω-regular sets of infinite words, Wagner was the first to discover remarkable relations between the two hierachies and completely describe them [Wag77]. The strictness of the Mostowski hierarchy for both deterministic and non-determinsitic automata on infinite trees was proved by Niwinski [Niw86] in 1986. The hierarchy problem for the alternating automata was solved by Bradfield [Brad98a] in 1996 (ten years after their introduction by Muller and Schupp). This was when he settled the hierarchy problem for the modal μ-calculus. For what concerns the Wadge hierarchy, Murlak gave in [Mur05] a full description of it for deterministic tree languages. In addition, he proved in [Mur08] that for deterministic languages the Wadge hierarchy is a huge refinement of the weak index hierarchy. It is therefore not unreasonable to conjecture that the same situation holds for ω-regular tree languages too. A first step in this direction was recently given by a nice work of Arnold and Niwinski. In [AN08] they have shown that, if we restrict our attention to the so-called game languages, the corresponding Wadge hierarchy is strict and its strictness implies the one of the Mostowski hierarchy of alternating tree automata. Unfortunately at the moment there is no description of the Wadge hierarchy for the classe of all recognizable tree languages[3].

Standard programming logics use *forward* modalities, which express weaker preconditions. However, there is a growing interests in enriching standard languages by adding *backward* modalities, expressing stronger postconditions. This kind of modalities motivates the study of procedures for the satisfiability problem of this class of languages. This endeavor is not easy. Mainly because the interaction between these two kinds of modalities can be quite tricky. For instance, even if modal μ-calculus with backwards modalities preserves the tree-model property, it loses the finite model one.

Two-way alternating tree automata were introduced by Vardi in [Var98] in order to prove the decidability of the satisfiability problem for the extension of the modal μ-calculus with backward modalities. These automata were also used by Grädel and Walukiewicz to solve the same problem but this time for guarded fixed point logic in [GW99]. As for the standard, or "one-way" case, it is natural to find an answer to the hierarchy problems for the two-way alternating tree automata. It is not difficult to see that Arnold's nice alternative proof of the strictness of Mostowski hierarchy for alternating tree automata can also be applied to the two-way case. Nevertheless, in this paper we would like to give another – very simple – solution to this problem. The idea is – following the approach of Arnold and Niwinski – to use Wadge games and the link between the Wadge hierarchy of game languages and the Mostowski hierarchy of two-way alternating automata. More precisely, we first obtain the hierarchy result by giving a very simple game theoretical proof of the strictness of the Wadge hierarchy of game languages – a result already proved in [AN08] – and then we prove that the strictness of this hierarchy implies the strictness of the Mostowski hierarchy of two-way alternating automata.

[3] Two first very partial attempts in this direction can be found in [DM07] and in [DF∞].

As we said, the main result can be obtained by applying the techniques presented in Arnold's proof of the strictness of the fixpoint hierarchy for the modal μ-calculus. Moreover we essentially make use of another already known result: the strictness of the Wadge hierarchy of game languages [AN08]. Nevertheless, this work is essentially methodological: it aims at stressing how much the Wadge games may be used in order to obtain easy proofs in hierarchical questions related to parity automata over trees.

2 Preliminaries

2.1 Alternating Tree Automata

Let W be a non empty alphabet. A tree over Σ is a partial function $t : W^* \to \Sigma$ with a prefix closed domain. Such trees may have infinite and finite branches. We call them *conciliatory.* A tree is said to be finitely branching when W is finite and binary if $W = \{1, 2\}$. Moreover, we call a tree *full* if it has only infinite branches. The elements of t are called nodes, and the empty word ε is the root of t. Let T_Σ denote the set of full binary trees over Σ and $T_\Sigma^{\leq\omega}$ denote the set of full finitely branching trees over Σ. Given $v \in \mathsf{dom}(t)$, by $t.v$ we denote the subtree of t rooted in v. In the sequel we only consider full binary trees over Σ. Thus, unless otherwise stated, when we speak of binary (resp. finitely branching) tree, we mean *full* binary (resp. finitely branching) tree.

For every $v \in \{1, 2\}^*$, $v.i$, where $i \in \{1, 2\}$, are the successors of v. If $i = 0$, then $v.i = v$, and for every $i \in \{1, 2\}$ we have $(v.i). - 1 = v$. If $v = \varepsilon$, then $v. - 1$ is undefined. If v is a node in the binary tree t different from the root, then $v. - 1$ denotes the (unique) ancestor of v.

Throughout the paper we will work with alternating automata running on full binary trees where the acceptance condition is given by a parity condition.

More precisely, an alternating parity tree automaton $\mathcal{A}$ is a tuple $\langle \Sigma, Q, \delta, (\iota, \kappa), \Omega \rangle$ where

- Σ is a finite non empty alphabet
- Q is a finite set of states
- δ is the transition function, it associates to every pair $(q, a) \in Q \times \Sigma$ an element of the free distributive lattice generated by $Q \times \{1, 2\}$,
- (ι, κ) is a pairs of natural numbers, called the *Mostowski index* of the automata, given as follows: $\iota \in \{0, 1\}$ and $\iota < \kappa \in \omega$,
- Ω is a mapping from Q into $\{\iota, \dots, \kappa\}$ such that for every $n \in \{\iota, \dots, \kappa\}$ there is a $q \in Q$ such that $\Omega(q) = n$.

Without loss of generality, we can suppose that for every $(q, a) \in Q \times \Sigma$, $\delta(q, a)$ is a finite non empty disjunction of finite non empty conjunctions. Thus, δ can be seen as a function from $Q \times \Sigma$ into $\wp(\wp(Q \times \{1, 2\}))$.

Let $t \in T_\Sigma$, and let $\mathcal{A}$ an alternating parity tree automata over the alphabet Σ defined as above. We define the parity game $\mathcal{P}(\mathcal{A}, t)$ as follows:

- the set of player 0's vertices is $V_0 = Q \times \{1, 2\}^*$

- the set of player 1's vertices is $V_1 = \wp(Q \times \{1,2\}) \times \{1,2\}^*$
- the initial position of the game is (q_I, ε),
- there is an edge from a node $(q, v) \in V_0$ to a node $(S, w) \in V_1$ iff $S \in \delta(q, t(v))$,
- there is an edge from a node $(S, v) \in V_1$ to a node $(q, v.i) \in V_0$, with $i \in \{1,2\}$ iff $(q, i) \in S$,
- a play $(q_1, v_1), (S_1, w_1), (q_2, v_2), (S_2, w_2), \dots$ is winning for player 0 iff the greatest priority occurring infinitely often in the sequence $\Omega(q_1), \Omega(q_2), \dots$ is even.

We say that the automaton $\mathcal{A}$ accepts the tree t if player 0 has a winning strategy in the parity game $\mathcal{P}(\mathcal{A}, t)$ starting from (q_I, ε). The language $L(\mathcal{A})$ accepted by the alternating parity tree automaton $\mathcal{A}$ over Σ is the subset of T_Σ of binary trees accepted by $\mathcal{A}$. Note that the arena of $\mathcal{P}(\mathcal{A}, t)$ is always a full finitely branching tree.

We consider the following partial order on indices of alternating automata:

$$(\iota, \kappa) \sqsubseteq (\iota', \kappa')$$

$$\text{iff}$$

$$\text{either } \{\iota, \dots, \kappa\} \subseteq \{\iota', \dots, \kappa'\} \text{ or } \{\iota + 2, \dots, \kappa + 2\} \subseteq \{\iota', \dots, \kappa'\}$$

The hierarchy induced by the partial order $\sqsubseteq$ on the class of alternating parity tree automata is called the *Mostowski hierarchy of alternating tree automata.*

Given the equivalence between formulae of the modal μ-calculus and alternating parity tree automata, it is not surprising that the Mostowski hierarchy corresponds to the fixpoint alternation hierarchy of the modal μ-calculus. By a result of Bradfield [Brad98a,Brad98b], we know that both hierarchies are strict.

2.2 Playing the Wadge Games

Consider the space $T_{\mathsf{B}}^{\leq\omega}$, equipped with the standard Cantor topology. Then, if $L, M \subseteq T_{\mathsf{B}}^{\leq\omega}$, we say that L is continuously reducible to M, if there exists a continuous function f such that $L = f^{-1}(M)$. We write $L \leq_w M$ iff L is continuously reducible to M. This is called the *Wadge ordering.* If $L \leq_w M$ and $M \leq_w L$, we write $L \equiv_w M$. If $L \leq_w M$ but not $M \leq_w L$, we write $L <_w M$. Thus, the Wadge hierarchy is the partial order induced by $<_w$ on the equivalence classes given by $\equiv_w$.

Let L and M be two arbitrary sets of full finitely branching trees. The Wadge game $\mathcal{W}(L, M)$ is played by two players, player I and player II. Both players build a tree, say t_{I} and t_{II}. At every round, player I plays first, both players add a level to their corresponding tree: they add children (at least one) to the terminal nodes of their corresponding tree. Player II is allowed to skip her turn, but not forever.

We say that player II wins the game iff $t_{\mathrm{I}} \in L \Leftrightarrow t_{\mathrm{II}} \in M$. This game is designed precisely in order to obtain:

Lemma 1 ([Wad84]). *Let $L, M \subseteq T_{\Sigma}^{\leq\omega}$. Then $L \leq_w M$ iff Player I has a winning strategy in the game $\mathcal{W}(L, M)$.*

2.3 Game Automata and Game Languages

Consider the alphabet $\Sigma_{(\iota,\kappa)} = \{0,1\} \times \{\iota, \dots, \kappa\}$ with $\iota \in \{0,1\}$ and $\iota \leq \kappa$. Then, to every full tree t over $\Sigma_{(\iota,\kappa)}$, we associate a parity game $\mathcal{P}(t)$ as follows: a node v in the tree is a position for player 0 iff the first component of the node is 0, and the rank of the node corresponds to its second component.

The set $W_{(\iota,\kappa)}$ corresponds to the class of trees in $T_{\Sigma_{(\iota,\kappa)}}$ for which Player 0 has a winning strategy in the corresponding parity game $\mathcal{P}(t)$. For every index (ι, κ), the set $W_{(\iota,\kappa)}$ is called the *the game language of index* (ι, κ).

The languages $W_{(\iota,\kappa)}$ are indeed ω-regular languages. More precisely, let $W_{(\iota,\kappa)}$ be the alternating tree automata $\mathcal{A}^{W}_{(\iota,\kappa)} = \langle \Sigma^W, Q^W, \delta^W, (\iota, \kappa), \Omega^W \rangle$ where:

- $\Sigma^W = \Sigma_{(\iota,\kappa)}$
- $Q^W = \{q_\iota, \dots, q_\kappa\}$
- for every $q \in Q^W$ and every $v \in \mathsf{dom}(t)$

$$\delta(q, v) = \begin{cases} \{\{(q_n, 1)\}, \{(q_n, 2)\}\} & \text{if } t(v) = (0, n), \\ \{\{(q_n, 1), (q_n, 2)\}\} & \text{if } t(v) = (1, n). \end{cases}$$

- $\Omega^W(q_n) = n$.

Then we have that $L(\mathcal{A}^{W}_{(\iota,\kappa)}) = W_{(\iota,\kappa)}$.

It is easy to see that every parity game $\mathcal{P}(\mathcal{A}, t)$ can be effectively encoded from the root into a full binary tree $t^W \in W_{(\iota,\kappa)}$, with (ι, κ) being the index of the automata $\mathcal{A}$, in such a way that Player 0 has a winning strategy in $\mathcal{P}(\mathcal{A}, t)$ iff she has a winning strategy in $\mathcal{P}(t^W)$. Moreover, this "encoding" is continuous. More precisely, we have the following:

Proposition 1. *Let $\mathcal{A}$ be any alternating parity tree automaton of index (ι, κ) over Σ. Then $L(\mathcal{A}) \leq_w W_{(\iota,\kappa)}$.*

Proof. Let $W'_{(\iota,\kappa)}$ be the class of all full finitely branching trees over $\Sigma_{(\iota,\kappa)}$ for which Player 0 has a winning strategy in the corresponding parity game $\mathcal{P}(t)$ Then, the winning strategy for Player II in the Wadge Game $\mathcal{W}(L(\mathcal{A}), W'_{(\iota,\kappa)})$ is given by just playing at every turn the arena of the parity game $\mathcal{P}(\mathcal{A}, t)$, where t is the finite tree precisely played by Player I at that moment. By Lemma 1 we have that $L(\mathcal{A}) \leq_w W_{(\iota,\kappa)}$. But clearly $W'_{(\iota,\kappa)} \leq_w W_{(\iota,\kappa)}$. Indeed, the winning strategy for Player II in the Wadge Game $\mathcal{W}(W'_{(\iota,\kappa)}, W_{(\iota,\kappa)})$ is the following. First, given the (finite) finitely branching tree $t : \{\epsilon, 0, 1, \dots, n\} \to \Sigma$, with $\mathsf{dom}(t) = \{\epsilon, 0, 1, \dots, n\}$, let $f(t) : \{0,1\}^* \to \Sigma$ be the *binary encoding* of t given by $f(t)(\varepsilon) = f(t)(1^j) = t(\varepsilon)$, with $1 \leq j < n$, $f(t)(1^n) = t(n)$ and $f(t)(1^j0) = t(j)$, for $0 \leq j < n$. We say that $\epsilon \in \mathsf{dom}(f(t))$ corresponds to $\epsilon \in \mathsf{dom}(t)$, $1^n \in \mathsf{dom}(f(t))$ corresponds to $n \in \mathsf{dom}(t)$ and that for every $1 \leq j < n$,

$1^j 0 \in \mathsf{dom}(f(t))$ corresponds to $j \in \mathsf{dom}(t)$. We then define inductively for every finite finitely branching tree t and for every node $v \in \mathsf{dom}(t)$, the binary encoding $f(t)$ of t and which is the (unique) node in $\mathsf{dom}(f(t))$ corresponding to v.

Consider the following strategy for Player II in $\mathcal{W}(W'_{(\iota,\kappa)}, W_{(\iota,\kappa)})$:

1. at the first round, copy Player I's move,
2. for every round $n \geq 2$, for every terminal node v of the tree constructed by Player I after round $n-1$, if t is the tree constructed after Player I's turn at round n, then replace the terminal node corresponding to v with the binary encoding of $t.v$.

By definition of the binary encoding, this is a well-defined *winning* strategy for Player II. By transitivity of $\leq_w$ we therefore obtain that $L(\mathcal{A}) \leq_w W_{(\iota,\kappa)}$. ■

It is worth noticing that the game languages witnesses the strictness of the Mostowski hierarchy of alternating tree automata[4]. This result was recently strengthened by Arnold and Niwinski in [AN08]. In their paper, the authors show that the class of game languages form a hierarchy with respect to the Wadge reducibility. In this section we give a very short proof of the the same results by exploiting the game-theoretical characterization of Wadge reducibility. But before this, we explicitely establish the link between the Wadge hierarchy and the Mostowski hierarchy.

Lemma 2. *If the Wadge hierarchy of tree game languages is strict, then the Mostowski hierarchy of alternating parity tree automata is strict too.*

Proof. Suppose that the Wadge hierararchy of for tree game languages is strict, but that the Mostowski index hierarchy for alternating parity tree automata collapses. Assume it collapses to the (ι, n) level. Consider the automata recognizing the game tree language $W_{(\iota,n+1)}$. By hypothesis, there is an alternating tree automaton $\mathcal{A}$ of index (ι, n) recognizing the same language. By Proposition 1, we are able to construct a winning strategy for Player I in the Wadge Game $\mathcal{W}(W_{(\iota,n+1)}, W_{(\iota,n)})$. But this contradicts the strictness of the Wadge hierarchy of tree game languages. ■

We now start to give another proof of Arnold and Niwinski's result. First, we have the following lemma:

Lemma 3. *For every n, Player I has a winning strategy in both $\mathcal{W}(W_{(1,n+1)}, W_{(0,n)})$ and $\mathcal{W}(W_{(0,n)}, W_{(1,n+1)})$.*

Proof. We only prove that player I has a winning strategy in $\mathcal{W}(W_{(1,n+1)}, W_{(0,n)})$, the other case being identical. We do this by describing the winning strategy for player I in this game. As a first move, player I plays a finite binary tree over $\{(1, 1)\}$. Then the strategy goes as follows:

[4] Cf. [Brad98b].

1. if player II skips, then for every terminal node add two children labelled by $(1,1)$
2. otherwise, at every terminal node, add two dual copies of what Player II has already played as successors.

Clearly this is a winning strategy for player I in $\mathcal{W}(W_{(1,n+1)}, W_{(0,n)})$. ■

Proposition 2 ([AN08]). *The Wadge hierarchy for tree game languages is strict.*

Proof. It is trivial to verify that, for every n, player II has a winning strategy in both $\mathcal{W}(W_{(0,n)}, W_{(0,n+1)})$ and $\mathcal{W}(W_{(0,n)}, W_{(1,n+2)})$, and dually in both $\mathcal{W}(W_{(1,n+1)}, W_{(1,n+2)})$ and $\mathcal{W}(W_{(1,n+1)}, W_{(0,n+1)})$. Therefore, if we show that player I has a winning strategy in $\mathcal{W}(W_{(0,n+1)}, W_{(0,n)})$ and $\mathcal{W}(W_{(1,n+2)}, W_{(0,n)})$ (and in the dual case) we are done. We only prove that player I has a winning strategy in $\mathcal{W}(W_{(0,n+1)}, W_{(0,n)})$, the other cases being identical. We do this by describing the winning strategy for player I in this game. As a first move, she plays an finite binary tree over the alphabet $\{(1,0)\}$. Then the strategy goes as follows:

1. if player II skips, then for every terminal node add two nodes labelled by $(1,0)$.
2. otherwise, by Lemma 3, apply the winning strategy for player I in the Wadge game $\mathcal{W}(W_{(1,n+1)}, W_{(0,n)})$.

Clearly this is a winning strategy for player I in $\mathcal{W}(W_{(0,n+1)}, W_{(0,n)})$. ■

By applying Proposition 2 to Lemma 2, we can therefore obtain, as a corollary, an alternative very easy proof of the strictness of the Mostowski hierarchy for alternating tree automata.

Corollary 1. *The Mostowski hierarchy for alternating parity tree automata is strict.*

Mutatis mutandis, the same argument yields the strictness of the Mostowski hierarchy of *weak* alternating parity automata.

2.4 Two-Way Alternating Parity Tree Automata

A two-way alternating parity automaton on binary trees is defined exactly as a (one-way) alternating atomata, except for the transition function and therefore for the parity game defining the acceptance condition. The transition function δ associates to every pair $(q,a) \in Q \times \Sigma$ an element of the free distributive lattice generated by $Q \times \{-1,0,1,2\}$. The associated paritiy game is defined as follows.

Let $t \in T_{\Sigma}$, and let $\mathcal{A}$ a two-way alternating parity tree automata over the alphabet Σ defined as above. Then the parity game $\mathcal{P}(\mathcal{A},t)$ is defined by the following conditions:

- the set of player 0 vertices is $V_0 = Q \times \{-1, 0, 1, 2\}^*$
- the set of player 1 vertices is $V_1 = \wp(Q \times \{-1, 0, 1, 2\}) \times \{-1, 0, 1, 2\}^*$
- the initial position of the game is (q_I, ε),
- there is an edge from a node $(q, v) \in V_0$ to a node $(S, w) \in V_1$ iff $S \in \delta(q, t(v))$,
- remember that if $i = 0$, then $v.i = v$, and that for every $i \in \{1, 2\}$ we have $(v.i). - 1 = v$. If $v = \varepsilon$, then $v. - 1$ is undefined. Thus, there is an edge from a node $(S, v) \in V_1$ to a node $(q, v.i) \in V_0$, with $i \in \{-1, 0, 1, 2\}$ iff (q, i) is defined and is a member of S,
- a play $(q_1, v_1), (S_1, w_1), (q_2, v_2), (S_2, w_2), \ldots$ is winning for player 0 iff the greatest priority occurring infinitely often in the sequence $\Omega(q_1), \Omega(q_2), \ldots$ is even.

As for the "one-way" case, we say that the automaton $\mathcal{A}$ accepts the tree t if player 0 has a winning strategy in the parity game $\mathcal{P}(\mathcal{A}, t)$ starting from (q_I, ε). The language $L(\mathcal{A})$ accepted by the to way alternating parity tree automaton $\mathcal{A}$ over Σ is the subset of T_Σ of binary trees accepted by $\mathcal{A}$. Note also that for two-way alternating parity automata, the arena of $\mathcal{P}(\mathcal{A}, t)$ is always a full finitely branching tree.

The hierarchy induced on the class of two-way alternating parity tree automata by the partial order $\sqsubseteq$ on indices of two-way automata is called the *Mostowski hierarchy of two-way alternating parity tree automata*.

This class of automata constitute the natural automata counterpart of the modal μ-calculus *with backward modalities*, that is the logic resulting from the addition of a universal backward modality and an existential backward modality to standard modal μ-calculus [Var98].

3 The strictness of the hierarchy

In order to solve the hierarchy problem for two-way alternating automata, we would like to apply the same argument as the one used in order to establish the strictness of the Mostowski hierarchy for one-way alternating automata over trees. Thence, as before we must establish a link between the Wadge hierarchy of game languages and the Mostowski hierarchy of two-way automata.

Remember that when constructing the parity game associated to the run of a two-way alternating automaton over a binary trees, the two new "backward" symbols -1 and 0 behave in fact as the usual "forward" symbols 1 and 2, that is to say the arena of the parity game a full finitely branching tree. Therefore, exactly as in the one-way case, we obtain the following proposition, whose proof is the same as for Proposition 1:

Proposition 3. *Let $\mathcal{A}$ be any two-way alternating parity tree automaton of index (ι, κ) over Σ. Then $L(\mathcal{A}) \leq_w W_{(\iota,\kappa)}$.*

Everything is now ready to prove the strictness of the Mostowski hierarchy for two-way alternating parity automata over trees. The remaining short step consists in establishing a relation between the Mostowski hierarchy and the Wadge hierarchy for game languages in the same way as it was done in the one-way case.

Lemma 4. *If the Wadge hierarchy of tree game languages is strict, then the Mostowski hierarchy of two-way alternating parity tree automata is strict too.*

Proof. Suppose that the Wadge hierararchy of tree game languages is strict, but that the Mostowski index hierarchy for two-way alternating parity tree automata collapses. Assume it collapses to the (ι, n) level. Consider the (two-way) automaton recognizing the game tree language $W_{(\iota,n+1)}$. By hypothesis, there is a two-way alternating tree automaton $\mathcal{A}$ of index (ι, n) recognizing the same language. By Proposition 3, we are able to construct a winning strategy for Player I in the Wadge Game $\mathcal{W}(W_{(\iota,n+1)}, W_{(\iota,n)})$. But this contradicts the strictness of the Wadge hierarchy of tree game languages. ■

Thus, by Lemma 4 and Proposition 2, we obtain the hierarchy result for two-way alternating parity automata:

Theorem 1. *The Mostowski hierarchy of two-way alternating parity tree automata is strict*

The correspondence between formulae of the modal μ-calculus with backward modalities and two-way alternating tree automata, also gives the strictness of the fixpoint hierarchy for this extension of the modal μ-calculus.

References

[Alb02] Alberucci L.: Strictness of the Modal μ-Calculus Hierarchy. In Grädel E., Thomas W., Wilke T. (Eds.): *Automata, Logics, and Infinite Games.* LNCS 2500, Springer-Verlag Berlin Heidelberg (2002), pp. 185–201

[Arn99] Arnold A. : The μ-Calculus Alternation-Depth Hierarchy is Strict on Binary Trees. *ITA* 33(4/5): 319-340 (1999)

[AN01] Arnold A., Niwinski D. : *Rudiments of μ-Calculus.* Studies in Logic vol. 146, Elsevier (2001)

[AN08] Arnold A., Niwinski D. : Continuous separation of game languages. *Fundamenta Informaticae*, 81(1-3): 19–28 (2008)

[Brad98a] Bradfield, J. : The Modal μ-Calculus Alternation Hierarchy is Strict. *Theor. Comput. Sci.* 195(1): 133–153 (1998)

[Brad98b] Bradfield, J. : Simplifying the Modal μ-Calculus Alternation Hierarchy. *STACS* 1998: 39–49 (1998)

[DF∞] Duparc J., Facchini A. : The Topological Complexity of Models of the Modal μ-Calculus – On The Alternation Free Fragment and Beyond (submitted)

[DM07] Duparc J., Murlak F. : On the Topological Complexity of Weakly Recognizable Tree Languages. *FCT 2007*, LNCS 4639: 261-273 (2007)

[Dup01] Duparc J. : Wadge Hierarchy and Veblen Hierarchy Part 1: Borel Sets of Finite Rank. *Journal of Symbolic Logic* 66(1): 56–86 (2001)

[EJ91] Emerson E.A., Jutla C.S. : Tree Automata, μ-Calculus and Determinacy (Extended Abstract). *FOCS* 1991: 368–377 (1991)

[GW99] Grädel E., Walukiewicz I.: Guarded fixed point logics. In *Proceedings 14th IEEE Symp. on Logic in Computer Science*, pp. 45–54 (1999)

[HKQ98] Henzinger T.A., Kupferman O., Qadeer S.: From prehistoric to postmodern symbolic model checking. *Proceedings of the Tenth International Conference on Computer-Aided Verification*, Lecture Notes in Computer Science 1427: 195–206 (1998)

[KSV96] Kupfermann O., Safra S., Vardi M. : Relating Word and Tree Automata. *1tth IEEE Symp. on Logic in Computer Science*: 322–332 (1996)

[Mos91] Mostowski A.W. : Hierarchies of weak automata and weak monadic formulas. *Theoret. Comput. Sci.* 83: 323–335 (1991)

[Mur05] Murlak F. : On deciding topological classes of deterministic tree languages. In *Proc. CSL '05*, Lecture Notes in Computer Science Vol. 3634 (2005): 573–584.

[Mur05] Murlak F. : The Wadge hierarchy of deterministic tree languages. In *Proc. ICALP '05*, Lecture Notes in Computer Science Vol. 4052 (2006): 408–419.

[Mur08] Murlak F. : Weak Index vs Borel Rank. *25th International Symposium on Theoretical Aspects of Computer Science (STACS 2008)*: 573–584 (2008)

[Niw86] Niwinski D. : On fixed point clones. In *Proc. ICALP '86*, Lecture Notes in Computer Science Vol. 226, Springer Verlag (1986): 464–473.

[NW05] Niwinski D., Walukiewicz I. : Deciding Nondeterministic Hierarchy of Deterministic Tree Automata. *Electr. Notes Theor. Comput. Sci.* 123: 195–208 (2005)

[Var98] Vardi M.Y. : Reasoning about The Past with Two-Way Automata. *Proceedings of the 25th International Colloquium on Automata, Languages and Programming*, Lecture Notes In Computer Science Vol. 1443: 628–641 (1998)

[Wad84] Wadge W.W.: *Reducibility and determinateness on the Baire space.* Ph.D. Thesis, Berkeley (1984)

[Wag77] Wagner K. : Eine topolgische Charackterisierung einiger Klassen regulärer Folgenmengen. *J. Inf. Process. Cybern.* EIK 13: 473–487 (1977)

[Wag79] Wagner K. : On ω-Regular stes. *Inform. and COntrol* 43: 123–177 (1979)

[Wil01] Wilke T.: Alternating Tree Automata, Parity Games, and Modal μ-Calculus. *Bul. Belg. Math. Soc.* 8(2): 359–391 (2001)