INFORMATIQUE THÉORIQUE ET APPLICATIONS

ANDRÉ ARNOLD

The μ-calculus alternation-depth hierarchy is strict on binary trees

Informatique théorique et applications, tome 33, n° 4-5 (1999), p. 329-339

<http://www.numdam.org/item?id=ITA_1999__33_4-5_329_0>

Theoretical Informatics and Applications
Theoret. Informatics Appl. **33** (1999) 329–339

THE μ-CALCULUS ALTERNATION-DEPTH HIERARCHY IS STRICT ON BINARY TREES

André Arnold[1]

Abstract. In this paper we give a simple proof that the alternation-depth hierarchy of the μ-calculus for binary trees is strict. The witnesses for this strictness are the automata that determine whether there is a winning strategy for the parity game played on a tree.

Résumé. Nous donnons dans cet article une preuve simple que la hiérarchie d'alternance de points fixes pour le μ-calcul sur les arbres binaires est infinie. Les témoins en sont les automates qui déterminent l'existence d'une stratégie gagnante pour le jeu de parité joué sur un arbre binaire.

AMS Subject Classification. 68Q68, 03D05.

Introduction

In the μ-calculus, the least and greatest fixed-point operators μ and ν are in some sense analogous to quantifiers in logics, and it is very natural to classify the formulas of the μ-calculus according to the alternation of these operators as it is done for logical formulas. This classification is called the alternation-depth hierarchy and its classes are denoted by Σ_n and Π_n.

Thus, a fundamental problem is the strictness of this hierarchy from the point of view of expressiveness: is it true or not that there exists an n such that any formula is equivalent to some formula in Σ_n. A necessary and sufficient condition for the strictness of the hierarchy is the existence for each n of a formula F_n that is in Σ_n but not in Π_n. Such a formula F_n is called Σ_n-hard.

When restricted to binary trees, the formulas of the μ-calculus are equivalent to alternating tree automata and the alternation-depth hierarchy corresponds to the Rabin index hierarchy of automata [15]. In 1986, Niwiński [14] showed that the

Keywords and phrases: Mu-calcul, alternating automata, parity games.

[1] LaBRI, Université de Bordeaux 1, CNRS, 351 cours de la Libération, 33405 Talence, France.

alternation hierarchy of the μ-calculus on binary trees *without intersection* (*i.e.*, the Rabin index hierarchy of non deterministic automata) is strict. But the fixed-point terms he considered for proving this result are all equivalent to co-Büchi terms (*i.e.* in Π_2) with intersection. Since then, the question of the strictness of the hierarchy of the μ-calculus on binary trees *with intersection* (*i.e.*, alternating automata) was open [4, 15].

Recently, and simultaneously, Bradfield [6] and Lenzi [10] have proved that the alternation-depth hierarchy of the modal μ-calculus is strict. In [7], Bradfield shows that the formula

$$B_n = \mu x_1.\nu x_2.\cdots.\theta x_n.[c]x_n \vee \langle a_1\rangle x_1 \vee \cdots \vee \langle a_n\rangle x_n$$

is Σ_n-hard, as well as the Walukiewicz's formula

$$W_n = \mu x_1.\nu x_2.\cdots.\theta x_n.\left(E \Rightarrow \langle\rangle \bigwedge_{i=1}^{n}(R_i \Rightarrow x_i)\right) \wedge \left(O \Rightarrow [] \bigwedge_{i=1}^{n}(R_i \Rightarrow x_i)\right),$$

that states that the first player has a winning strategy in a parity game [17] and that is nothing but the extension of the Emerson-Jutla's formula [8] to games that are not bipartite.

The Lenzi's Π_n- and Σ_n-hard formulas are formulas on n-ary trees. They are defined inductively by

$$\begin{aligned} L_0(P) = L_0^d(P) &= P, \\ L_{n+1}(P) &= \nu x_{n+1}.(P \wedge L_n^d(a_{n+1}.x_{n+1})), \\ L_{n+1}^d(P) &= \mu x_{n+1}.(P \vee L_n(a_{n+1}.x_{n+1})), \end{aligned}$$

where a node has property $a_i.x$ if its ith successor has property x.

Since Lenzi's formulas are formulas on n-ary trees, and since one can encode n-ary trees into binary trees, one can deduce from Lenzi's result that the alternation hierarchy of the μ-calculus is also strict on binary trees. Such a transformation is not so easy for Bradfield's and Walukiewicz's formulas [5].

In this note we offer a direct proof that Walukiewicz's formulas are hard on binary trees. It combines two arguments already used by Bradfield in [7].

The *reduction argument* is that every Σ_n-formula F reduces to the Walukiewicz's formula W_n via some mapping G_F, *i.e.*, $M \models F \Leftrightarrow G_F(M) \models W_n$. The *diagonal argument* is just the fact that any mapping G_F has a fixed point M_F. Thus, if the negation of the Walukiewicz's formula W_n is equivalent to a Σ_n-formula F, we get $M_F \models F \Leftrightarrow M_F \models \neg F$, a contradiction. The existence of this fixed point is a consequence of the celebrated Banach's fixed-point theorem, since the mappings G_F are contracting on the compact metric space of binary trees.

It turns out that the same diagonal argument can be applied to *weak alternating automata* introduced by Muller *et al.* to characterize the weakly definable sets of trees [12], providing a direct proof of the Mostowski's result on the hierarchy of weak alternating automata [11], instead of relying on a result of Thomas [16].

1. Alternating parity automata

An alternating parity automaton is an alternating automaton (see [13]) where the acceptance criterion is given by a parity condition. Namely, it is a tuple $\langle A, Q, \delta, n, r\rangle$ where

- the *alphabet* A is a finite set of binary symbols,
- Q is a finite set of states,
- n is a natural number ($n > 0$), called the *type* of $\mathcal{A}$ and r is a mapping from Q to $\{1, \dots, n\}$,
- δ associates with each $q \in Q$ and each $a \in A$ an element of the free distributive lattice generated by $Q \times \{1,2\}$.

Indeed, each $\delta(q,a)$ can be seen as a finite disjunction of finite conjunctions of elements in $Q \times \{1,2\}$. Without loss of generality, we may assume that $\delta(q,a)$ is a finite *non empty* disjunction of finite *non empty* conjunctions.

Example: Walukiewicz's automata.
Let A_n be the alphabet $\{c_i, d_i \mid 1 \leq i \leq n\}$, and let T_n be the set of binary trees over A_n (*i.e.*, the set of mappings $t : \{1,2\}^* \to A_n$). The Walukiewicz automaton $\mathcal{W}_n$ is $\langle A_n, Q_n, \delta_n, n, r_n\rangle$ where

- $Q_n = \{q_1, \dots, q_n\}$ and for any $q_i \in Q_n$, $r_n(q_i) = i$,
- for any $q \in Q$, $\delta(q, c_i) = (q_i, 1) \wedge (q_i, 2)$ and $\delta(q, d_i) = (q_i, 1) \vee (q_i, 2)$.

The set $R_{\mathcal{A},q}(t)$ of *runs* of $\mathcal{A}$ from the state q on a tree t built on the alphabet A is the set of (unordered) trees, labeled by states in Q, defined recursively as follows: $\rho \in R_{\mathcal{A},q}(a(t_1,t_2))$ if and only if

- the root of ρ is labeled by q,
- among the conjunctions that are added up to form $\delta(q,a)$, there exists a conjunction $(q_1, x_1) \wedge \cdots \wedge (q_k, x_k)$ such that the root of ρ has exactly k subtrees that are respectively in $R_{\mathcal{A},q_j}(t_{x_j})$ ($j = 1, \dots, k$).

A branch b of such a run ρ (that is always infinite because of our assumption) is *μ-accepting* (resp. *ν-accepting*) if the minimum value of $r(q)$ where q ranges over the set of states that occur infinitely often on b is even (resp. odd).

A run ρ is *μ-accepting* (resp. *ν-accepting*) if each of its branches is μ-accepting (resp. ν-accepting). Finally $L^{\mu}_{\mathcal{A}}(q)$ (resp. $L^{\nu}_{\mathcal{A}}(q)$) is the set of all trees t such that $R_{\mathcal{A},q}(t)$ contains at least one μ-accepting run (resp. ν-accepting).

Example: Walukiewicz's tree languages.
In an automaton $\mathcal{W}_n$ we have, by definition, $\delta_n(q_i, a) = \delta_n(q_j, a)$ for any $a \in A_n$ and any $i, j \in \{1, \dots, n\}$. It follows that for $\theta = \mu, \nu$ and for any i, j, $L^{\theta}_{\mathcal{W}_n}(q_i)$ $= L^{\theta}_{\mathcal{W}_n}(q_i)$. We denote this language by W^{θ}_n. It is interesting to notice that the intersections of these W^{θ}_n with the set of trees over the alphabet $\{c_i \mid i = 1, \dots, n\}$

are exactly the tree languages defined by Niwiński in [14] to show the strictness of the hierarchy of non deterministic automata.

The trees that belongs to W_n^θ can be characterized in terms of game. Let t be a tree in T_n. A node $u \in \{1,2\}^*$ is a C-node (resp. D-node) if $t(u) \in \{c_1, \dots, c_n\}$ (resp. $\{d_1, \dots, d_n\}$). Let two players C and D. Initially, a token is at the root ε of t. If the token is at C-node (resp. D-node) u then player C (resp. D) moves the token to node $u1$ or $u2$. A play is an infinite word of $\{1,2\}^\omega$, *i.e.*, a branch of t, and D wins the play if the least index that occurs infinitely often on this branch is even or odd, according to the value of θ. A strategy s for D is a mapping that associates $s(u) \in \{1,1\}$ with each D-node u. The strategy s is winning if D wins any play consistent with this strategy, *i.e.*, at the D-node u, D moves the token to $us(u)$. What the Walukiewicz automata does is just to guess a strategy (by the rules $\delta(q,d_i) = (q_i,1) \vee (q_i,2)$) and accepts t if the strategy it guessed is winning. Therefore $t \in W_n^\theta$ if and only if D has a winning strategy in the game on t.

Let $\widetilde{\mathcal{A}}$ be the automaton obtained from $\mathcal{A}$ by exchanging $\vee$ and $\wedge$. The complementation theorem of Muller and Schupp [13] now reads as follows, where T_A is the set of all binary trees built on the alphabet A.

Proposition 1.1. $L^\mu_{\widetilde{\mathcal{A}}}(q) = T_A - L^\nu_{\mathcal{A}}(q), \quad L^\nu_{\widetilde{\mathcal{A}}}(q) = T_A - L^\mu_{\mathcal{A}}(q)$.

Definition 1.2. We denote by Σ_n (resp. Π_n) the family of binary tree languages in the form $L^\mu_{\mathcal{A}}(q)$ (resp. $L^\nu_{\mathcal{A}}(q)$) for some automaton $\mathcal{A}$ of type n. In particular, $W_n^\mu \in \Sigma_n$ and $W_n^\nu \in \Pi_n$.

As a consequence of the previous proposition, we get

Proposition 1.3. *For any tree language L over the alphabet A,*

$$L \in \Sigma_n \Leftrightarrow T_A - L \in \Pi_n.$$

Definition 1.4. We say that a language L is Σ_n-*hard* if it is in Σ_n and not in Π_n (or, equivalently, its complement $T_A - L$ is not in Σ_n).

2. The reduction argument

It is well-known (see [9] and [8], for instance) that the acceptance of a tree t by an automaton $\mathcal{A}$ can be expressed as the existence of a winning strategy in a game G associated with $\mathcal{A}$ and t. When $\mathcal{A}$ is a parity automaton, the game G is a parity game and the existence of a (memoryless) winning strategy is expressed by the Walukiewicz's formulas. The same argument is used by Bradfield [7] to show hardness of Walukiewicz's formulas. We show that when $\mathcal{A}$ is of rank n, we can construct an associated game that is indeed a binary tree in T_n, and the existence of a winning strategy is asserted by the membership of this tree to a Walukiewicz's language.

Let $\mathcal{A}$ be an automaton of rank n over the alphabet A. For any state q of $\mathcal{A}$, we define recursively the mapping $G_{\mathcal{A},q} : T_A \to T_n$ as follows.

We can see $\delta(q,a)$ as a finite binary tree whose internal nodes are labelled by $\vee$ and $\wedge$ and leaves by elements of $Q \times \{1,2\}$. Then $G_{\mathcal{A},q}(a(t_1,t_2))$ is the tree obtained by substituting in $\delta(q,a)$

- c_i for $\wedge$, d_i for $\vee$, where $i = r(q)$,
- the tree $G_{\mathcal{A},q'}(t_x)$ for (q',x).

The following characterization is nothing but another way of explaining when a tree is accepted by a parity automaton, like in [8], or, in other words, the winning strategy for D in the game on $G_{\mathcal{A},q}(t)$ allows us to construct an accepting run in $R_{\mathcal{A},q}(t)$, and conversely, from an accepting run, we can construct a winning strategy.

Proposition 2.1. *For $\theta = \mu,\nu$ and $t \in T_A$, $t \in L^\theta_{\mathcal{A}}(q) \Leftrightarrow G_{\mathcal{A},q}(t) \in W^\theta_n$.*

3. The diagonal argument

Since, in an automaton $\mathcal{A}$, substituting $\delta(q,a) \vee \delta(q,a)$ for $\delta(q,a)$ does not modify the set of runs, we may assume that each tree $\delta(q,a)$ has its root labelled by $\vee$. Therefore, if we consider that T_A and T_n are equipped with the usual ultrametric distance Δ defined by $\Delta(t,t') \leq 2^{-k} \Leftrightarrow \forall u \in \{1,2\}^*$, $|u| < k \Rightarrow t(u) = t'(u)$, that makes T_A and T_n complete, and even compact [2], it is easy to see that, for any automaton $\mathcal{A}$ and any state q, the mapping $G_{\mathcal{A},q} : T_A \to T_n$ is contracting, provided the above assumption on each $\delta(a,q)$.

Proposition 3.1. $\Delta(G_{\mathcal{A},q}(t), G_{\mathcal{A},q}(t')) \leq \Delta(t,t')/2$.

Proof. It is easy to prove by induction on k that $\Delta(t,t') \leq 2^{-k} \Rightarrow \Delta(G_{\mathcal{A},q}(t), G_{\mathcal{A},q}(t')) \leq 2^{-(k+1)}$.

In particular, if $\mathcal{A}$ is an automaton of rank n over the alphabet T_n, the contracting mapping $G_{\mathcal{A},q} : T_n \to T_n$ has a unique fixed point $t_{\mathcal{A},q}$ and we get, from Proposition 2.1,

Proposition 3.2. $t_{\mathcal{A},q} \in L^\theta_q(\mathcal{A}) \Leftrightarrow t_{\mathcal{A},q} \in W^\theta_n$.

Theorem 3.3. *The Walukiewicz's language W^μ_n is Σ_n-hard.*

Proof. Clearly $W^\mu_n \in \Sigma_n$. Now, assume that $T_n - W^\mu_n \in \Sigma_n$. Then there exists an automaton $\mathcal{A}$ of rank n and a state q such that $T_n - W^\mu_n = L^\mu_q(\mathcal{A})$ and we get $t_{\mathcal{A},q} \in T_n - W^\mu_n \Leftrightarrow t_{\mathcal{A},q} \in W^\mu_n$, an obvious contradiction. Hence $W^\mu_n \in \Sigma_n - \Pi_n$. By a similar reasoning, we get $W^\nu_n \in \Pi_n - \Sigma_n$. □

4. Universal languages

The previous diagonal argument can be easily extended to prove the hardness of some tree languages.

We say that a language L over an alphabet A is Σ_n*-universal* if there is a non expansive mapping $F : T_n \to T_A$ [1] such that

$$\forall t \in T_n, \quad t \in W_n^\mu \Leftrightarrow F(t) \in L.$$

Note that we do not require L to be in Σ_n.

Similarly, we say that L is Π_n*-universal* if

$$\forall t \in T_n, \quad t \in W_n^\nu \Leftrightarrow F(t) \in L.$$

In particular, taking F as the identity mapping over T_n, we get that W_n^μ is Σ_n-universal and that W_n^ν is Π_n-universal.

Theorem 4.1. *A Σ_n-universal language is never in Π_n. A Π_n-universal language is never in Σ_n.*

Proof. Let L be a Σ_n-universal language. If it is in Π_n, $\overline{L} = T_A - L$ is in Σ_n, and by the reduction argument (Prop. 2.1) there exists a contracting mapping $G_{\overline{L}} : T_A \to T_n$ such that $t \notin L \Leftrightarrow G_{\overline{L}}(t) \in W_n^\mu$. Since L is Σ_n-universal, $G_{\overline{L}}(t) \in W_n^\mu \Leftrightarrow F(G_{\overline{L}}(t)) \in L$. Since $G_{\overline{L}}$ is contracting and F is non expansive, $F \circ G_{\overline{L}} : T_A \to T_A$ is contracting and has a fixed point $t_{\overline{L}}$ that satisfies $t_{\overline{L}} \notin L \Leftrightarrow t_{\overline{L}} \in L$, a contradiction.

The second part of the theorem is proved quite similarly.

Corollary 4.2. *If a Σ_n-universal language is in Σ_n then it is Σ_n-hard.*

Example: Bradfield's tree languages.
Let $\mathcal{B}_n$ be the automaton over the alphabet $A'_n = \{c_n\} \cup \{d_i \mid 1 \leq i \leq n\}$ defined by

- $Q = \{q_1, \dots, q_n\}$ and for any $q_i \in Q_n$, $r(q_i) = i$,
- for any $q \in Q$, $\delta(q, c_n) = (q_n, 1) \wedge (q_n, 2)$ and $\delta(q, d_i) = (q_i, 1) \vee (q_i, 2)$.

Because of the analogy between these automata and the Bradfield's formulas, we call them *Bradfield's automata.* The *Bradfield's languages* are the languages $B_n^\theta = L_{q_1}^\theta(\mathcal{B}_n)$ that are in Σ_n or Π_n according to whether θ is μ or ν. Indeed it is easy to see that $B_n^\theta = W_n^\theta \cap T'_n$ where T'_n is the set of all binary trees over the alphabet $A'_n \subseteq A_n$.

Proposition 4.3. *B_n^μ is Σ_n-universal. B_n^ν is Π_n-universal.*

Proof. Let $F : T_n \to T'_n$ be defined by

- F is the identity on A'_n,
- for $i < n$, $F(c_i(x_1, x_2)) = c_n(d_i(x_1, x_1), d_i(x_2, x_2))$.

We establish a correspondence between the runs of $\mathcal{W}_n$ on t and the runs of $\mathcal{B}_n$ on $F(t)$ as follows.

- We apply the rule $(q, d_i(t_1, t_2)) \to (q_i, t_j)$ in $\mathcal{W}_n$ if and only if we apply the rule $(q, d_i(F(t_1), F(t_2))) \to (q_i, F(t_j))$ in $\mathcal{B}_n$ (with $j = 1$ or 2).

[1] F is non expansive if $\forall t, t', \Delta(F(t), F(t')) \leq \Delta(t, t')$.

- We apply the rule $(q, c_n(t_1, t_2)) \to (q_n, t_1) \wedge (q_n, t_2)$ in $\mathcal{W}_n$ if and only if we apply the rule $(q, c_n(F(t_1), F(t_2))) \to (q_n, F(t_1)) \wedge (q_n, F(t_2))$ in $\mathcal{B}_n$.
- We apply the rule $(q, c_i(t_1, t_2)) \to (q_i, t_1) \wedge (q_i, t_2)$ in $\mathcal{W}_n$ (with $i < n$) if and only if we apply in $\mathcal{B}_n$ the derivation

$$\begin{aligned} & (q, c_n(d_i(F(t_1), F(t_1)), d_i(F(t_2), F(t_2)))) \\ \to \quad & (q_n, d_i(F(t_1), F(t_1))) \wedge (q_n, d_i(F(t_2), F(t_2))) \\ \to \quad & (q_i, F(t_1)) \wedge (q_i, F(t_2)). \end{aligned}$$

This correspondence preserves the set of states that appears infinitely often on any branch, except that in $\mathcal{B}_n$ we may add infinitely often q_n. But since $r(q_n) = n$, the minimal rank of these sets remains unchanged. Therefore, one of two corresponding runs is θ-accepting if and only if so is the other one. Since B_n^μ is in Σ_n, we get, by Corollary 4.2:

Corollary 4.4. *B_n^μ is Σ_n-hard.*

Example: Lenzi's tree languages.

The Lenzi's formulas are formulas over n-ary trees. Translating these formulas into alternating parity automata over binary trees that encode n-ary trees, we get the following definition of Lenzi's automata $\mathcal{L}_n$, for $n \geq 2$.

The alphabet of $\mathcal{L}_n$ is $A_n'' = \{a_i \mid 0 \leq i \leq n\}$ and we denote by T_n'' the set of all trees over this alphabet. Its set of states is $\{q_i \mid 1 \leq i \leq n\}$ with $r(q_i) = i$, and three additional states $\{q_0, q_a, q_r\}$ such that

- q_0 accepts only trees in the form $a_0(t_1, t_2)$, its rank does not matter since it will occur at most once on any branch of a run,
- q_a, of rank 2, accepts any tree,
- q_r, of rank 1, accepts no tree.

It is not difficult to write down the rules implementing these conditions.

The other rules are, for $i = 1, \ldots, n$,

$$\delta(q_i, a_i) = \begin{cases} (q_{i+1}, 1) \vee (q_{i-1}, 2) & \text{if } i \text{ is odd,} \\ (q_{i+1}, 1) \wedge (q_{i-1}, 2) & \text{if } i \text{ is even,} \end{cases}$$

where we assume that $q_{i+1} = q_n$ whenever $i \geq n$.

When $i \neq j$ the rule $\delta(q_j, a_i)$ is not defined. We assume that in this case the automaton rejects the tree, *i.e.* $\delta(q_j, a_j) = (q_r, 1) \wedge (q_r, 2)$.

Let $L_n^\theta = L_{q_1}^\theta(\mathcal{L}_n)$.

The following proposition shows that $L_{n+2}^\mu \in \Sigma_{n+2} - \Pi_n$ and that $L_{n+2}^\mu \in \Pi_{n+2} - \Sigma_n$. They are not exactly hard in the sense above (*i.e.*, in $\Sigma_{n+2} - \Pi_{n+2}$ or in $\Pi_{n+2} - \Sigma_{n+2}$). However they also provide evidence for the strictness of the alternation hierarchy: if the hierarchy is not strict, there exists n such that $\Pi_n = \Sigma_n = \Sigma_{n+2}$.

Proposition 4.5. *L^{μ}_{n+2} is Σ_n-universal. L^{ν}_{n+2} is Π_n-universal.*

Proof. We recursively define a family of non expansive mappings $F_i : T_n \to T''_{n+2}$, for $i = 1, \ldots, n+2$, such that $t \in W^{\theta}_n \Leftrightarrow F_i(t) \in L^{\theta}_i(\mathcal{L}_{n+2})$, and we take $F = F_1$.

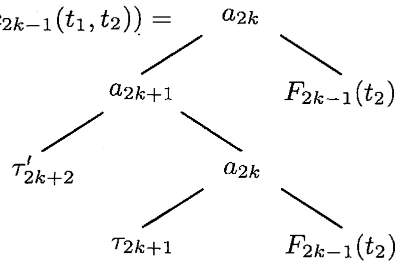
First we show that for each $i = 0, \ldots, n+2$ there exist a tree $\tau_i \in L^{\theta}_i(\mathcal{L}_{n+2})$ and a tree $\tau'_i \notin L^{\theta}_i(\mathcal{L}_{n+2})$. Obviously, τ'_i is an arbitrary tree whose the root is not a_i. τ_0 is an arbitrary tree whose the root is a_0, $\tau_{2i+1} = a_{2i+1}(\tau_{2i}, \tau_{2i})$, and τ_{2i+2} is the unique tree t such that

$$t = \begin{cases} a_{2i+2}(t, \tau_{2i+1}) & \text{if } 2i+2 = n+2, \\ a_{2i+2}(a_{2i+3}(\tau'_{2i+4}, t), \tau_{2i+1}) & \text{otherwise,} \end{cases}$$

with the convention, that will also be used in the sequel, that if an index becomes strictly greater than $n+2$ it has to be understood as having the value $n+2$, or in other words, each index i has to be read as $\min(i, n+2)$.

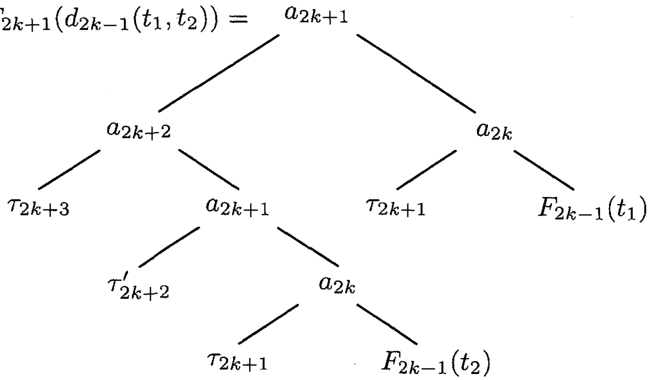
Now, for $2k-1 \leq n$, we define

$F_{2k}(c_{2k-1}(t_1, t_2)) =$



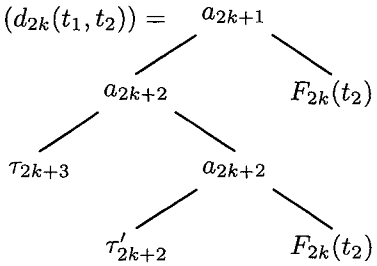
and

$F_{2k+1}(d_{2k-1}(t_1, t_2)) =$



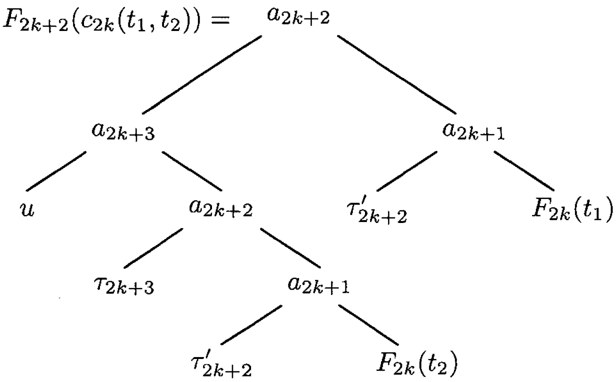
For $2k \leq n$, we define

$F_{2k+1}(d_{2k}(t_1, t_2)) =$



and

$F_{2k+2}(c_{2k}(t_1, t_2)) =$



where $u = \begin{cases} \tau_{n+2} & \text{if } 2k+2 = n+2, \\ \tau'_{2k+4} & \text{otherwise.} \end{cases}$

Now, for $T \in T_n$, let

$$\gamma(t) = \begin{cases} 2k+2 & \text{if the root of } t \text{ is } c_{2k} \text{ or } d_{2k+1}, \\ 2k+1 & \text{if the root of } t \text{ is } c_{2k-1} \text{ or } d_{2k}. \end{cases}$$

We have just defined $F_i(t)$ for $i = \gamma(t)$. For $i \neq \gamma(t)$, we set

$$F_{2j}(t) = \begin{cases} a_{2j}(\tau_{2j+1}, F_{2j-1}(t)) & \text{if } 2j > \gamma(t), \\ a_{2j}(F_{2j+1}(t), \tau_{2j-1}) & \text{if } 2j < \gamma(t), \end{cases}$$

$$F_{2j+1}(t) = \begin{cases} a_{2j+1}(\tau'_{2j+2}, F_{2j}(t)) & \text{if } 2j+1 > \gamma(t), \\ a_{2j+1}(F_{2j+2}(t), \tau'_{2j}) & \text{if } 2j+1 < \gamma(t). \end{cases}$$

We prove that $t \in L^\theta_{q_i}(\mathcal{W}_n) = W^\theta_n \Leftrightarrow F_i(t) \in L^\theta_{q_i}(\mathcal{L}_n)$. We remark that $F_i(b_k(t_1, t_2))$, for $b_k = c_k, d_k$, has the form $t(F_k(t_1), F_k(t_2))$. It is easy to see that an accepting run from state q_i on $F_i(t)$ is made up of accepting runs from q_j on the subtrees τ_j, of an accepting run from q_k on $F_k(t_1)$ and (or) on $F_k(t_2)$, and that on the path from the root to $F_k(t_1)$ and (or) $F_k(t_2)$, the only states that appears are q_i, q_k and some q_j with $j \geq \min(i, k)$.

5. Weak alternating automata

An alternating automaton $\mathcal{A} = \langle A, Q, \delta, n, r\rangle$ is said to be *weak* if δ has the following additionnal property:

for all $q \in Q$ and for all $a \in A$, if q' occurs in $\delta(q, a)$, then $r(q') \leq r(q)$.

It is obvious that if $\mathcal{A}$ is weak, then its dual $\widetilde{\mathcal{A}}$ is weak too.

The *weak alternation-depth hierarchy* is defined in the same way as the alternation depth hierarchy: $w\Sigma_n$ (resp. $w\Pi_n$) is the family of binary tree languages in the form $L^\mu_{\mathcal{A}}(q)$ (resp. $L^\nu_{\mathcal{A}}(q)$) where $\mathcal{A}$ is a weak automaton of type n. Since the family of languages accepted by weak alternating automata is exactly the family of languages L accepted by non deterministic Büchi automata whose the complement is also accepted by a non deterministic Büchi automaton, and since the family of languages accepted by a non deterministic Büchi automaton is exactly Σ_2 [1, 3], we have

$$\bigcup_{n>0} (w\Sigma_n \cup w\Pi_n) = \Sigma_2 \cap \Pi_2.$$

A direct consequence of the definition of a weak automaton $\mathcal{A}$ is that on any branch of $G_{\mathcal{A}}(q, t)$ the sequence of indices of the letters c_i, d_i $(i = 1, \dots, n)$ occurring on this branch is decreasing. Therefore, in such a case, Proposition 2.1 can be restated as:

$$\text{for } \theta = \mu, \nu \text{ and } t \in T_A,\ t \in L^\theta_{\mathcal{A}}(q) \Leftrightarrow G_{\mathcal{A},q}(t) \in L^\mu_{w\mathcal{W}_n}(q_n),$$

where $w\mathcal{W}_n$ is a variant of the Walukiewicz's automaton $\mathcal{W}_n$ that takes into account this property of decreasing indices, *i.e.*,

- $\delta(q_j, c_i) = (q_i, 1) \wedge (q_i, 2)$ and $\delta(q_j, d_i) = (q_i, 1) \vee (q_i, 2)$ if $j \geq i$.
- $\delta(q_j, c_i) = \delta(q_j, d_i) = (q_1, 1) \vee (q_1, 2)$ if $j < i$.

Obviously, $w\mathcal{W}_n$ is weak, and the above diagonal construction allows us to prove that $L^\mu_{w\mathcal{W}_n}(q_n)$ is not in $w\Pi_n$ (resp. $L^\nu_{w\mathcal{W}_n}(q_n)$ is not in $w\Sigma_n$). This is a new proof of a Mostowski's result [11]

Theorem 5.1 (Mostowski). *The alternation-depth hierarchy of weak automata is strict.*

REFERENCES

[1] A. Arnold, Logical definability of fixed points. *Theoret. Comput. Sci.* **61** (1988) 289–297.

[2] A. Arnold and M. Nivat, The metric space of infinite trees. Algebraic and topological properties. *Fund. Inform.* **4** (1980) 445–476.

[3] A. Arnold and D. Niwiński, Fixed-point characterization of büchi automata on infinite trees. *J. Inf. Process. Cybern. EIK* **26** (1990).

[4] A. Arnold and D. Niwiński, Fixed point characterization of weak monadic logic definable sets of trees, M. Nivat and A. Podelski, Eds., *Tree automata and Languages.* Elsevier (1992) 159–188.

[5] J.C. Bradfield, Fixpoint alternation: Arithmetic, transition systems, and the binary tree, *this issue.*

[6] J.C. Bradfield, The modal mu-calculus alternation hierarchy is strict, U. Montanari and V. Sassone, Eds., in *Proc. CONCUR '96, Lecture Notes in Comput. Sci.* **1119** (1996) 233–246.

[7] J.C. Bradfield, Simplifying the modal mu-calculus alternation hierarchy, M. Morvan, C. Meinel and D. Krob, Eds., in *Proc. STACS '98, Lecture Notes in Comput. Sci.* **1373** (1998) 39–49.

[8] E.A. Emerson and C.S. Jutla, Tree automata, mu-calculus and determinacy, in *Proc. FOCS '91.* IEEE Computer Society Press (1991) 368–377.

[9] Y. Gurevitch and L. Harrington, Trees, automata and games, in *Proc. 14th ACM Symp. on the Theory of Computing* (1982) 60–65.

[10] G. Lenzi, A hierarchy theorem for the mu-calculus, F. Meyer auf der Heide and B. Monien, Eds., in *Proc. ICALP '96, Lecture Notes in Comput. Sci.* **1099** (1996) 87–109.

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

[12] D.E. Muller, A. Saoudi and P.E. Schupp, Alternating automata, the weak monadic theory of the tree and its complexity. *Theoret. Comput. Sci.* **97** (1992) 233–244.

[13] D.E. Muller and P.E. Schupp, Alternating automata on infinite trees, *Theoret. Comput. Sci.* **54** (1987) 267–276.

[14] D. Niwiński, On fixed point clones, L. Kott, Ed., in *Proc. 13th ICALP, Lecture Notes in Comput. Sci.* **226** (1986) 464–473.

[15] D. Niwiński, Fixed points characterization of infinite behaviour of finite state systems. *Theoret. Comput. Sci.* **189** (1997) 1–69.

[16] W. Thomas, A hierarchy of sets of infinite trees, A.B. Cremers and H.P. Kriegel, Eds., *Theoret. Comput. Sci., Lecture Notes in Comput. Sci.* **145** (1983) 335–342.

[17] I. Walukiewicz, Monadic second-order logic on tree-like structures, C. Puech and R. Reischuk, Eds., in *Proc. STACS '96, Lecture Notes in Comput. Sci.* **1046** (1996) 401–414.

Received November 5, 1998. Revised May 1, 1999.