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

André Arnold*

In 1986, Niwiński [13] showed that the alternation hierarchy of the μ-calculus on binary trees *without intersection* was strict. But the fixed-point terms he considered for proving this result are all equivalent to co-Büchi terms (i.e. in Σ_2) with intersection.

Since then, the question of the strictness of the hierarchy of the μ-calculus on binary trees *with intersection* was open [4, 14].

Recently, and simultaneously, Bradfield [5] and Lenzi [9] have proved that the alternation hierarchy of the modal μ-calculus is strict. In [6], Bradfield shows that the formula

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

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

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

that states that the first player has a winning strategy in a parity game [16] and that is nothing but the extension of the Emerson-Jutla's formula [7] 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.

In this note we offer a direct proof that Walukiewicz's formulas are hard on binary trees. It combines two arguments used by Bradfield in [6]:

*LaBRI, Université Bordeaux I and CNRS (UMR 5800)

- a *reduction argument*: a formula F reduces to a formula F' if for any pointed graph M, there exists a pointed graph $G(M)$ such that $M \models F \Leftrightarrow G(M) \models F'$.
- a *diagonal argument*: there is a Σ_n-formula that expresses satisfiability of any Σ_n-formula.

Our proof uses the same reduction argument as Bradfield's one: every Σ_n-formula F reduces to a Walukiewicz's formula via some mapping G_F. The diagonal argument is just the fact that any mapping G_F has a fixed point M_F, thus if the negation of a Walukiewicz's formula is equivalent to a Σ_n-formula F, we get $M_F \models F \Leftrightarrow M_F \models \neg F$. 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.

Indeed, since we are concerned with binary trees, we use the equivalence between formulas of the μ-calculus on trees and parity automata stated by Niwiński in [14]: with every formula F is associated an alternating parity automaton $\mathcal{A}$, and vice-versa, such that a tree is a model of F if and only if it is accepted by $\mathcal{A}$.

It turns out that the same diagonal argument can be applied to *weak alternating automata* introduced by Muller, Saoudi, and Schupp to characterize the weakly definable sets of trees [11], providing a direct proof of the Mostowski's result on the hierarchy of alternating automata [10], instead of relying on a result of Thomas [15].

1 Alternating parity automata

An alternating parity automaton is an alternating automaton (see [12]) 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

- $Q_n = \{q_1, \ldots, q_n\}$ 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 $R_{\mathcal{A},q_j}(t_{x_j})$ $(j = 1, \ldots, k)$.

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 $i, j \in \{1, \ldots, n\}$. 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 intersection of these W^{θ}_n with the set of trees over the alphabet $\{c_i \mid i = 1, \ldots, n\}$ are exactly the tree languages defined by Niwiński in [13] to show the strictness of the hierarchy of non deterministic automata.

Let $\widetilde{\mathcal{A}}$ be the automaton obtained from $\mathcal{A}$ by exchanging $\vee$ and $\wedge$. The complementation theorem of Muller and Schupp [12] now reads as follows, where T_A is the set of all binary trees built on the alphabet A.

Proposition 1 $L^{\mu}_{\widetilde{\mathcal{A}}}(q) = T_A - L^{\nu}_{\mathcal{A}}(q), \quad L^{\nu}_{\widetilde{\mathcal{A}}}(q) = T_A - L^{\mu}_{\mathcal{A}}(q).$

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, $W^{\mu}_n \in \Sigma_n$ and $W^{\nu}_n \in \Pi_n$.

As a consequence of the previous proposition, we get

Proposition 2 *For any tree language L over the alphabet A, $L \in \Sigma_n \Leftrightarrow T_A - L \in \Pi_n$.*

2 The reduction argument

It is well-known (see [8] and [7], 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 [6] 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 [7].

Proposition 3 *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 4 $\Delta(G_{\mathcal{A},q}(t), G_{\mathcal{A},q}(t')) \leq \Delta(t,t')/2$.

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 3,

Proposition 5 $t_{\mathcal{A},q} \in L^\theta_q(\mathcal{A}) \Leftrightarrow t_{\mathcal{A},q} \in W^\theta_n$.

Now, assume that $W^\mu_n \in \Sigma_n$ is also in Π_n, i.e., $T_n - W^\mu_n \in \Sigma_n$. 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 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, $T_A - L$ is in Σ_n, and by the reduction argument (Proposition 3) there exists a contracting mapping $G_L : T_A \to T_n$ such that $t \notin L \Leftrightarrow G_L(t) \in W_n^\mu$. Since L is Σ_n-universal, $G_L(t) \in W_n^\mu \Leftrightarrow F(G_L(t)) \in L$. Since G_L is contracting and F is non expansive, $F \circ G_L : T_A \to T_A$ is contracting and has a fixed point t_L that satisfies $t_L \notin L \Leftrightarrow t_L \in L$, a contradiction.

The second part of the theorem is proved quite similarly.

Example: Bradfield's tree languages

Let $\mathcal{B}_n$ be the automaton over the alphabet $A'_n = \{c_n\} \cup \{d_i \mid 1 \le i \le n\}$ defined by

- $Q = \{q_1, \ldots, q_n\}$ 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 6 *B_n^μ is Σ_n-universal. B_n^ν is Π_n-universal.*

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

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

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

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.

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 $n + 1 = n$.

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.

Proposition 7 *L_{n+2}^μ is Σ_n-universal. L_{n+2}^ν 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 $t \in W_n^\theta \Leftrightarrow F_i(t) \in L_i^\theta(\mathcal{L}_{n+2})$, and we take $F = F_1$.

First we show that for each $i = 0, \ldots, n+2$ there exist a tree $\tau_i \in L_i^\theta(\mathcal{L}_{n+2})$ and a tree $\tau'_i \notin L_i^\theta(\mathcal{L}_{n+2})$. 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

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

with the convention that $n+3 = n+2$.

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

$$F_{2k}(c_{2k-1}(t_1, t_2)) = a_{2k}\Big(a_{2k+1}\big(\tau'_{2k+2},\ a_{2k}(\tau_{2k+1}, F_{2k-1}(t_2))\big),\ F_{2k-1}(t_2)\Big)$$

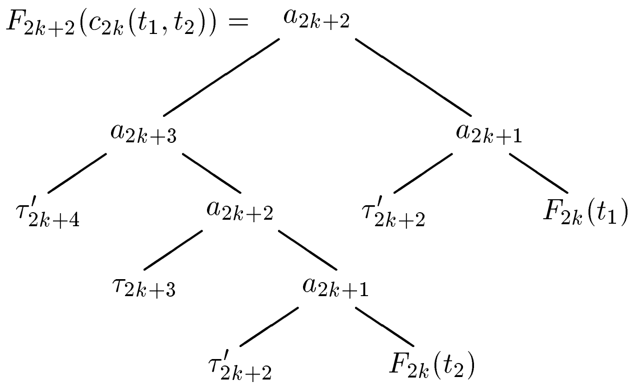
and

$$F_{2k+1}(d_{2k-1}(t_1, t_2)) = a_{2k+1}\Big(a_{2k+2}\big(\tau_{2k+3},\ a_{2k+1}(\tau'_{2k+2}, a_{2k}(\tau_{2k+1}, F_{2k-1}(t_2)))\big),\ a_{2k}(\tau_{2k+1}, F_{2k-1}(t_1))\Big)$$

For $2k \leq n$, we define

$$F_{2k+1}(d_{2k}(t_1, t_2)) = a_{2k+1}\Big(a_{2k+2}\big(\tau_{2k+3},\ a_{2k+2}(\tau'_{2k+2}, F_{2k}(t_2))\big),\ F_{2k}(t_2)\Big)$$

always assuming that $n+3=n+2$, and



where we still assume that $n+3=n+2$ and, moreover, that if $2k=n$, τ_{n+2} is substituted for τ'_{2k+4}.

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

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

where $n+3=n+2$.

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 ($i = 1, \ldots, n$) occuring on this branch is decreasing. Therefore, in such a case, Proposition 3 can be restated as:

For $\theta = \mu, \nu$ 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 $w\Sigma_n$). Of course, it is also possible to define the $w\Sigma_n$- and $w\Pi_n$-universal languages.

References

[1] A. Arnold. Logical definability of fixed points. *Theoretical Computer Science*, 61:289–297, 1988.

[2] A. Arnold and M. Nivat. The metric space of infinite trees. Algebraic and topological properties. *Fundamenta Informaticae*, 4:445–476, 1980.

[3] A. Arnold and D. Niwiński. Fixed-point characterization of büchi automata on infinite trees. *J. Inf. Process. Cybern. EIK*, 26:453–461, 1990.

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

[5] J. C. Bradfield. The modal mu-calculus alternation hierarchy is strict. In U. Montanari and V. Sassone, editors, *Proc. CONCUR '96*, pages 233–246. Lect. Notes Comput. Sci. 1119, 1996.

[6] J. C. Bradfield. Simplifying the modal mu-calculus alternation hierarchy. In M. Morvan, C. Meinel, and D. Krob, editors, *Proc. STACS '98*, pages 39–49. Lect. Notes Comput. Sci. 1373, 1998.

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

[8] Y. Gurevitch and L. Harrington. Trees, automata and games. In *Proc. 14th ACM Symp. on the Theory of Computing*, pages 60–65, 1982.

[9] G. Lenzi. A hierarchy theorem for the mu-calculus. In F. Meyer auf der Heide and B. Monien, editors, *Proc. ICALP '96*, pages 87–109. Lect. Notes Comput. Sci. 1099, 1996.

[10] A. W. Mostowski. Hierarchies of weak automata and weak monadic formulas. *Theoretical Computer Science*, 83:323–335, 1991.

[11] D. E. Muller, A. Saoudi, and P. E. Schupp. Alternating automata and a weak monadic theory of the tree and its complexity. In L. Kott, editor, *Proc. 13th ICALP*, pages 275–283. Lect. Notes Comput. Sci. 226, 1986.

[12] D. E. Muller and P. E. Schupp. Alternating automata on infinite trees. *Theoretical Computer Science*, 54:267–276, 1987.

[13] D. Niwiński. On fixed point clones. In L. Kott, editor, *Proc. 13th ICALP*, pages 464–473. Lect. Notes Comput. Sci. 226, 1986.

[14] D. Niwiński. Fixed points characterization of infinite behaviour of finite state systems. *Theoretical Computer Science*, 189:1–69, 1997.

[15] W. Thomas. A hierarchy of sets of infinite trees. In A. B. Cremers and H. P. Kriegel, editors, *Theoretical Computer Science Proc.*, pages 335–342. Lect. Notes Comput. Sci. 145, 1983.

[16] I. Walukiewicz. Monadic second-order logic on tree-like structures. In C. Puech and R. Reischuk, editors, *Proc. STACS '96*, pages 401–414. Lect. Notes Comput. Sci. 1046, 1996.