Theoretical Computer Science 83 (1991) 323-335
Elsevier

Note

Hierarchies of weak automata and weak monadic formulas

A.W. Mostowski

Institute of Mathematics, University of Gdańsk, Wita Stwosza 57, 80-952 Gdańsk, Poland

Communicated by A. Salomaa
Received March 1990

Mostowski, A.M., Hierarchies of weak automata and weak monadic formulas (Note), Theoretical Computer Science 83 (1991) 323-335.

The paper deals with alternating finite automata (a.f.a.) on trees. They are investigated in [6, 5, 7]. Rabin indices and Muller indices for automata, which are well known in relevant literature, are a natural measure of complexity of the automata (Definition 1.1). Muller index (or Rabin index) of a set of trees is a minimal index of a Muller (or Rabin) automaton which represents this set. In this paper it is proved that both indices, Rabin and Muller, are equal for weak as well as for strong conditions (Theorem 1.4). For the classical automata the equality of indices for strong conditions was proved in [3]; the infinity of the hierarchy in [8].

The main result (Theorems 5.1-5.3) states that: for k successor monadic arithmetic, a weak formula with a bounded suffix has m unbounded quantifiers in the prefix iff it is represented by a weak a.f.a. of index m, except for $m=0$ and $m=1$ in a case when some weak variables occur freely in a formula. Then the representing weak a.f.a. must have index 2.

From this result and from Thomas' result [13] on infinity of weak quantifier's hierarchy, it follows that the hierarchy of weak indices for a.f.a. is infinite.

The paper uses neither Rabin's representation theorem [10] nor the determinancy theorem for games [2]. The result on complements of a.f.a., proved in [7] or [6], suffices to prove our results similarly as it suffices to prove the result in [5] on representing weak formulas by weak Muller a.f.a. and vice versa.

Section 1 is introductory on a.f.a. and its indices. Equality of indices is stated here but proved in Section 2 which is devoted to a.f.a. with Rabin conditions, weak as well as strong. For weak a.f.a., useful constructions of a subset automaton and of a projection automaton are described in Section 3. Section 4 is on weak formulas. Finally, Section 5 is devoted to a proof of the main results.

1. Alternating automata

We suppose that the reader is familiar with a philosophy of a.f.a. as exposed in [5, 6, 7]. A table of an a.f.a. M is of a form $\langle L(K \times U), \Sigma, \delta, u0\rangle$ where Σ is an alphabet, K a set of directions of the tree, U a set of states and $u0 \in U$ an initial state. The sets are supposed to be finite.

The δ is a transition function giving for any $u \in U$ and $x \in \Sigma$ an element $\delta = \delta(u, x)$ of a free distributive lattice $L(K \times U)$ generated by $K \times U$. Then δ is uniquely represented as $\delta = \delta_1 \vee \cdots \vee \delta_s$ where

$$\delta_i = (k1 \times u1) \wedge \cdots \wedge (kr \times ur) \tag{1}$$

for $i = 1, \ldots, s$. Here $kj \in K$ and $uj \in U$ for $j = 1, \ldots, r$. They (and the r) depend on the i.

Function $t: K^* \to \Sigma$ is called an (infinite) Σ-tree.

At the beginning of an acceptation process of a tree by an a.f.a. M, a single copy of M starts in the root of t with the initial state $u0$. During this process a copy of M, being in a state u, in a node labelled by x, "chooses" a summand of $\delta(u, x)$, i.e. a δi given by (1), and "produces" its copies, e.g. $M1, \ldots, Mr$ sending them in the directions: $k1, \ldots, kr$ in the states $u1, \ldots, ur$, respectively. Either many or no copies can be sent in a certain direction. The process is taken step by step. At infinity each copy has its history

$$h = u0\, u1\, u2 \ldots . \tag{2}$$

The acceptance condition is a set of successful histories. For Muller automata they are described by an $\mathbb{F} \subseteq P(U)$, but for Rabin automata by a set $\Gamma = \{(L_{2i}, L_{2i-1}), i = 1, \ldots, I\}$, where $L_1, \ldots, L_{2I} \subseteq U$. We shall explain this is more detail. First some definitions. For a history h define:

$$u \in \mathrm{St}(h) \text{ iff } u = uj \quad \text{for some } j = 0, 1, \ldots \text{ in (2)},$$

$$u \in \mathrm{Inf}(h) \text{ iff } u = uj \quad \text{for infinitely many } j = 0, 1, \ldots \text{ in (2)}.$$

In the sequel we shall often deal with weak conditions defined in [5]. Then classical conditions will here be named strong conditions.

For weak conditions, a history satisfies $\mathbb{F}$ iff $\mathrm{St}(h) \in \mathbb{F}$, a history satisfies Γ iff for some $i = 1, \ldots, I$,

$$\mathrm{St}(h) \cap L_{2i} \neq \emptyset \quad \text{and} \quad \mathrm{St}(h) \cap L_{2i-1} = \emptyset.$$

Note that for a weak Rabin automaton, satisfying a pair, e.g. (L, L'), means visiting L at least once and not visiting L' at all.

For strong conditions Inf is used in the place of St.

Definition 1.1. An a.f.a. M consists of its table $\langle L(K \times U), \Sigma, \delta, u0\rangle$ and an acceptance condition, described by $\mathbb{F}$ (Muller a.f.a.) or Γ (Rabin a.f.a.), either weak or strong. A Σ-tree t is accepted by an a.f.a. M, which we shall write $t \vdash M$ iff an acceptation process of M on t exists, such that all histories of copies appearing during the process are successful. A language $L(M)$ represented by M consists of all Σ-trees t such that $t \vdash M$.

An index of an automaton is defined in the same way for weak as well as for strong conditions.

Definition 1.2. Index of Γ is a number of different sets in $L_1, \ldots, L_{2I}$ excluding the sets: $\emptyset$ and U. Index of $\mathbb{F}$ is a maximal number of changes $X_i \in \mathbb{F}$, $X_{i+1} \notin \mathbb{F}$, or vice versa in an ascending chain

$$\emptyset = X_1 \subset X_2 \subset \cdots \subset X_z = U.$$

Now we can define an index for a set of trees.

Definition 1.3. For a class of automata weak or strong, either Muller or Rabin, a minimal index of an automaton M representing the language L is called the index of L. If no such automaton exists the index is supposed to be infinite.

According to the definition we have two weak Muller and Rabin indices and two strong indices, also Muller's and Rabin's. The following holds.

Theorem 1.4. *For any index weak or strong, the Rabin and the Muller indices of any language of Σ-trees are equal.*

Proof. (a) The inequality Rabin index $\geqslant$ Muller index. The Γ gives a set $\mathbb{F} = \{X$: there exists $i = 1, \ldots, I$ such that $X \cap L_{2i} \neq \emptyset$ and $X \cap L_{2i-1} = \emptyset\}$. Then a Rabin automaton for Γ is a Muller automaton for $\mathbb{F}$. Now if $X1 < X2 < X3$ and $Xi \cap L \neq \emptyset$ and $Xi \cap L' = \emptyset$ for $i = 1$ and $i = 3$, then also for $i = 2$. Hence the inequality Rabin index $\geqslant$ Muller index holds for weak as well as for strong indices.

(b) The inequality Rabin index $\leqslant$ Muller index. It is far from being evident. The proof follows directly from Theorem 2.2 for weak as well for strong indices. □

Let us note that for classical automata, the equality of both types of strong indices is proved in [3] and the infinity of the hierarchy of indices is proved in [8].

2. Rabin automata

In this section we shall discuss the equivalence problems for Rabin and Muller automata.

Definition 2.1. Let a chain of subsets of a set Q be given

$$\emptyset \neq L_1 \subset L_2 \subset \cdots \subset L_m = Q. \tag{3}$$

A Rabin a.f.a. with the set of states Q is structured for (3) if the following hold:

(a) The transition function δ is such that when a copy of M reaches a state from L_i for some $i = 1, \ldots, m$, then all its new born copies remain in states from the L_i for ever.

(b) The conditions are either Γ or Γ' where

$$\Gamma = \{(L_2, L_1), \ldots, (L_{2\lfloor m/2 \rfloor}, L_{2\lfloor m/2 \rfloor - 1})\},$$
$$\Gamma' = \{(L_1, \emptyset), \ldots, (L_{2\lfloor (m-1)/2 \rfloor + 1}, L_{2\lfloor (m-1)/2 \rfloor})\}. \tag{4}$$

Note that according to the definition, $\mathrm{index}(\Gamma) = \mathrm{index}(\Gamma') = m - 1$. Moreover a history h is successful for Γ iff it is unsuccessful for Γ'. Then by the complementation theorem (see [5-7]), the automata $M = \langle L(K \times Q), \Sigma, \delta, q0, \Gamma \rangle$ and $M' = \langle L(K \times Q), \Sigma, \delta', q0, \Gamma' \rangle$ where δ' is the dual of δ in $L(K \times Q)$ are complementary for weak as well as for strong conditions given by Γ or Γ'.

For proving the equivalence of Muller and Rabin a.f.a., and for proving the equality of the indices for sets of trees, it remains to prove the following.

Theorem 2.2. *For any weak (resp. strong) Muller a.f.a.*

$$N = \langle L(K \times U), \Sigma, \delta 1, u0, \mathbb{F} \rangle,$$

a weak (resp. strong) Rabin a.f.a.

$$M = \langle L(K \times Q), \Sigma, \delta, q0, \Omega \rangle$$

can be effectively constructed such that a set Q of states is graduated by (3); *the conditions Ω are either Γ or Γ' given by* (4); $\mathrm{index}(\Omega) = m - 1 = \mathrm{index}(\mathbb{F})$ *where m is given by* (3). *Moreover for weak a.f.a. N, the automaton M is structured by* (3). *If N has n states then, for weak a.f.a., M has less than $n2^n$ states; for strong a.f.a., M has less than $mn2^{2^n}$ states.*

Proof. Preliminaries: Set $\mathbb{F}_0 = \mathbb{F} \setminus \{\emptyset\}$ and for $i = 0, 1, 2, \ldots,$

$$\mathbb{X}_i = \{X : X \subset U, P(X) \subset \mathbb{F}_i\}, \qquad \mathbb{F}_{i+1} = (P(U) \setminus \mathbb{F}_i) \cup \mathbb{X}_i.$$

Then $\emptyset = \mathbb{X}_0 \subset \mathbb{X}_1 \subset \mathbb{X}_2 \subset \cdots \subset \mathbb{X}_m = P(U)$ for some m, and $\mathbb{F} \setminus \{\emptyset\} = \bigcup_{i=1,2,\ldots} (\mathbb{X}_{2i} \setminus \mathbb{X}_{2i-1})$.

Surely $\mathrm{index}(\mathbb{F}) \leq m - 1$ since no $X, X' \in \mathbb{X}_{2i} \setminus \mathbb{X}_{2i-1}$ cause a change. A proof that $m - 1 \leq \mathrm{index}(\mathbb{F})$ is by induction on $|\mathbb{F}|$.

(A) Now we shall construct a simulating automaton for weak conditions. Set $Q=\{u\times s\colon u\in U,\ s\in P(U),\ u\in s\}$. Then we define $\delta(q,x)$ as follows: for $q=u\times s$, $u\in s$, $x\in\Sigma$ and $\delta 1(u,x)=\delta 1_1\vee\cdots\vee\delta 1_s$ and $\delta 1_j=(k1\times u1)\wedge\cdots\wedge(kr\times ur)$, define for $j=1,\dots,s$, the summand $\delta_j=\delta_j(q,x)=(k1\times(u1\times s1))\wedge\cdots\wedge(kr\times(ur\times sr))$ where $si=s\cup\{ui\}$ for $i=1,\dots,r$, next define $\delta(q,x)=\delta_1\vee\cdots\vee\delta_s$. Note that the states reached by $\delta 1$ are remembered on the s coordinate.

It remains to define the graduation

$$\emptyset=L_0\subset L_1\subset L_2\subset\cdots\subset L_m=Q.$$

Set

$$q=u\times s\in L_i \text{ iff } q\in Q \text{ and } s\in P(U)\backslash\mathbb{X}_{m-i}.$$

Now for $U\in\mathbb{F}$ set $\Omega=\Gamma'$ and for $U\notin\mathbb{F}$ set $\Omega=\Gamma$ where Γ, Γ' are given by (4). Also set $q0=u0\times\{u0\}$.

By a definition of δ there is one to one correspondence between the acceptation processes of N and those of the simulating automaton M. Now let $h1=u0\,u1\,u2\dots$ be a history of N and let $h=(u0\times s0)\,(u1\times s1)\,(u3\times s3)\dots$ be a history of the simulating automaton M. Then by the definition of δ we have $si=\{u0,\dots,ui\}$ for $i=0,1,2,\dots$.

For these histories there are some moments, e.g. $t1, t2,\dots$ such that $sj\in\mathbb{X}_{i+1}\backslash\mathbb{X}_i$ for $j\geqslant ti$ which in turn means, by the definition of a gradation on Q, that $qj=uj\times sj\in L_{m-i}$.

This proves that M really simulates the action of N and is equivalent to N. The automaton M is structured. The part (a) of Definition 2.1 follows from the inclusions $si\subseteq s(i+1)$ for $i=0,1,2,\dots$ and part (b) follows from the definition of Ω.

(B) The construction of a simulating automaton for strong conditions is based on the previous construction.

We define $Q=\{q\colon q=u\times w\times R$ where $w\subseteq P(U)$, $u\in s$ for each $s\in w$, and R is some extra record$\}$. For a $x\in\Sigma$ and a $q=u\times\{s1,\dots,st)\times R$, where $u\in si$ for $i=1,\dots,t$ and $\delta 1(u,x)=\delta 1_1\vee\cdots\vee\delta 1_s$ and $\delta 1_j=(k1\times u1)\wedge\cdots\wedge(kr\times ur)$ define for $j=1,\dots,s$, the summand $\delta_j=\delta_j(q,x)=(k1\times(u1\times w1\times R1))\wedge\cdots\wedge(kr\times(ur\times wr\times Rr))$ where $wi=\{\{ui\},s1\cup\{ui\},\dots,st\cup\{ui\}\}$ for $i=1,\dots,r$. Then we define $\delta(q,x)=\delta_1\vee\cdots\vee\delta_s$.

We can see that in each moment we simply start a new simulating automaton for weak conditions which gathers the states reached from this moment. All these automata have a common steering in a choice of $\delta 1_j$, so they all simulate an action of N.

Set

$$q0=u0\times\{u0\}\times R0$$

where $R0$ is the j such that $\{u0\}\in\mathbb{X}_j$ and $\{u0\}\notin\mathbb{X}_{j-1}$.

For a full definition of the δ there remains to define the records $R1,\dots,Rr$.

Let q and q' be two consecutive states of a copy of the simulating automaton M; e.g. $q=u\times\{s1,\dots,st)\times R$ and $q'=u'\times\{s0\cup\{u'\},\ s1\cup\{u'\},\dots,st\cup\{u'\}\}\times R'$,

$s0 = \emptyset$. We shall say that there is a switch on j iff for some $p = 0, 1, \ldots, t$, $sp \notin \mathbb{X}_j$ but $sp \cup \{u'\} \in \mathbb{X}_j$. Then R' is the biggest j such that there is a switch on j. Note that R' does not depend on R.

Now let us define the graduation

$$q = u \times w \times R \in L_{m-j} \text{ iff } R > j.$$

Then

$$\emptyset = L_0 \subset L_1 \subset L_2 \subset \cdots \subset L_m = Q.$$

Let now $h1 = u0\, u1\, u2 \ldots$ be a history of N. Then $\inf(h1) = F$ iff for some i: $\mathrm{st}(un\, u(n+1) \ldots) = F$ for all $n \geq i$. Hence if $F \in \mathbb{X}_i \setminus \mathbb{X}_{i-1}$ then for the simulating automaton there is a switch on $j > i$ only finitely many times and a switch on $j = i$ infinitely many times.

Now for $U \in \mathbb{F}$ set $\Omega = \Gamma'$ and for $U \notin \mathbb{F}$ set $\Omega = \Gamma$ where Γ, Γ' are given by (4). Then the automaton M really simulates N.

Now to finish the proof there remains to see that the estimations concerning the number of states follows from the constructions given in (A) and (B). □

3. Subset construction and projection automata

Here we shall give two very useful constructions for weak automata. All automata investigated in this and in the next sections will be weak.

If we wish one copy of an automaton at most to exist in a node we must group all copies $M1, \ldots, Mn$, of the a.f.a. M, existing in a node in states $q1, \ldots, qn$ into one supercopy SM existing in the node in a state $\{q1, \ldots, qn\}$. A formal definition of transitions of SM is somewhat complicated by the fact that two copies, e.g. M' and M'' may exist in the same state e.g. q, but they use different transitions, e.g. $\delta_{i'}$ and $\delta_{i''}$ from

$$\delta = \delta_1 \vee \cdots \vee \delta_{i'} \vee \cdots \vee \delta_{i''} \vee \cdots \vee \delta_s = \delta(q, x).$$

Now let us define $\delta^+ = \delta \vee (\delta \wedge \delta) \vee (\delta \wedge \delta \wedge \delta) \vee \cdots$. The expression is in fact finite. Then $\delta_{i'} \wedge \delta_{i''}$ is a summand in δ^+. For a $u = \{q1, \ldots, qn\}$, $n > 0$ define $\delta 1 = \delta 1(u, x) = \delta^+(q1, x) \wedge \cdots \wedge \delta^+(qn, x)$. Both in δ^+ and in $\delta 1$ let us present the elements as a sum of products using only distributivity, associativity, commutativity and idempotency, but without using absorption laws. Then all choices of transitions for all copies can be realised by one choice for $\delta 1$; it remains only to group the resulting copies into a supercopy.

For a product $a = (l1 \times q1') \wedge \cdots \wedge (l1 \times qu') \wedge (l2 \times q(u+1)') \wedge \cdots \wedge (lr \times q(v+1)') \wedge \cdots \wedge (lr \times qw')$, we define $\mathrm{group}(a) = (l1 \times \{q1', \ldots, qu'\}) \wedge \cdots \wedge (lr \times \{q(v+1)', \ldots, qw'\})$. For a sum we define $\mathrm{group}(a \vee b) = \mathrm{group}(a) \vee \mathrm{group}(b)$.

Now for the automaton SM let us define a transition function $\delta(u, x)$ for $u \in Q' = P(Q) \setminus \{\emptyset\}$ and $x \in \Sigma$ as follows:

$$\delta(u, x) = \mathrm{group}(\delta 1(u, x)).$$

Definition 3.1. An automaton $SM = \langle L(K \times Q'), \Sigma, \delta, \{q0\}\rangle$ with δ as above, and with unspecified conditions is called a subset automaton.

By adding a state $\emptyset$ it can be extended to a classical (i.e. nonalternating) automaton acting on trees. By a run, the automaton SM simulates to some extent the behaviour of a "bunch of processes" of M. The following lemma explains the extent of this simulation.

Lemma 3.2. *Let $D = (Q, W)$ be a Rabin pair and let the automaton M be structured for $\emptyset \subset W \subset Q$. Then in an acceptation process of M a D-history (i.e. a history not reaching a state from W) exists if and only if in the corresponding run of the SM there exist infinitely many states $u \in P(Q)$ such that $u \cap W \neq \emptyset$.*

Proof. The only if part is immediate. Now suppose that in the run there exists infinitely many states such that $u \cap W \neq \emptyset$. For any such state, any $q \in u$ is owed to some copy of M, with a previous history, e.g. $q0\, q1 \ldots qn$, where $qn = q$. Since the automaton is structured $q \notin W$, it follows that all $q0, q1, \ldots, qn \notin W$. Since the set of histories of copies appearing in an acceptation process of M is compact and complete, there must be a copy of M with an infinite D-history. □

Let now $\Sigma \supset \Sigma'$ be alphabets and η a projection of Σ onto Σ'. A Σ'-tree language L' is a finite tree projection η of a language L of Σ-trees, if for any $t' \in L'$ there exists a $t \in L$, and a finite tree $\Delta \subset K^*$ such that $t' = \eta t$ and for any $y \in K^*$, $y \in \Delta$ iff $t(y) \notin \Sigma'$.

Our aim is to construct an automaton PM accepting a finite tree projection of a set L represented by a weak Rabin a.f.a. M.

Let $M = \langle L(K \times Q), \Sigma, \delta, q0, \Omega\rangle$. Let $Q2$ be a disjoint sum of Q and $Q' = P(Q)\backslash\{\emptyset\}$. Hence q and $\{q\}$ must be carefully distinguished. Define

$$\delta 2(q, x') = \delta(q, x') \quad \text{for } q \in Q \text{ and } x' \in \Sigma',$$

$$\delta 2(u, x') = [\bigvee \delta(u, x)] \vee [\bigwedge \delta(q, x')] \quad \text{for } u \in Q' \text{ and } x' \in \Sigma'. \tag{5}$$

The sum in the first square brackets is for all $x \in \Sigma$ such that $x\eta = x'$ and $x \notin \Sigma'$, and gives a "guessing" part of the behaviour of PM. The δ is the δ from SM. The product in the second square brackets is for all $q \in u$ and gives a "splitting" part of the behaviour of PM.

Definition 3.3. An automaton $PM = \langle L(K \times Q2), \Sigma', \delta 2, \{q0\}, \Omega\rangle$ is called a projection automaton for Σ, Σ' and M. Note that Ω is the same as in M.

Note that if M is structured for (3) then PM is structured for $\emptyset \neq L_1 \subset L_2 \subset \cdots \subset L_m = Q \subset L_{m+1} = Q2$. Indeed if we reach Q then the transitions are possible only by the first line of (5), and do not leave Q.

In the beginning PM acts as a classical automaton on a Σ'-tree t', and guesses an $x \notin \Sigma'$ as long as a choice of a summand is made from the first square brackets part of the second line of (5). An action of M on a Δ part of a guessed Σ-tree t is simulated. The moment a choice is made from the second square brackets part of the second line of (5) then leaving of the Δ is anticipated, the "splitting" is done and all previously grouped copies start to live independently. Above the Δ the automaton starts to act as M does using the first line of (5) for transitions. The guessed tree is such that $t\eta = t'$. The conditions ensure that splitting must be always done in some moment since a history remaining entirely inside Q' is unsuccessful. This gives the following.

Lemma 3.4. *If M represents a set L of Σ-trees then* PM *represents a finite tree projection by η of L.*

The lemma is very similar to Lemma 2 from [5]. The only reason we make such a complicated construction is our need to estimate indices. For Γ, Γ' as in (4): for $\Omega = \Gamma$, m odd and for $\Omega = \Gamma'$, m even $\mathrm{index}(\mathrm{PM}) = \mathrm{index}(M)$; for $\Omega = \Gamma'$, m odd and for $\Omega = \Gamma$, m even $\mathrm{index}(\mathrm{PM}) = 1 + \mathrm{index}(M)$.

4. Weak theory of a tree

In a weak monadic theory of a tree there are two kinds of variables: $y, z, \ldots$ ranging over elements of K^*, and finite set variables $\alpha, \beta, \ldots$ ranging over finite subsets of K^*.

For a collection, e.g. $P_1, \ldots, P_v$, of unary predicates on K^*, and a collection $\{\alpha\}$ of finite set variables, we can choose the atomic formulas as αy, $P_i y$, $y = zk$, $y = z$ for any individual variables y and z, any $\alpha \in \{\alpha\}$, any $k \in K$ and $i = 1, \ldots, v$. Here αy means $y \in \alpha$, Py means $P(y)$ and $y = zk$ means y is the k successor of z. Using constants in place of individual variables is also allowed. The use of $\emptyset y = \mathrm{false}$ is also allowed.

Quantification over P is not allowed in a weak theory. The other predicates sometimes used in the definition of a monadic theory of a tree can be defined in our theory. Especially $\alpha \leqslant \beta \overset{\mathrm{def}}{=} \forall y\, (\alpha y \Rightarrow \exists z\, (y \leqslant z \,\&\, \beta z))$ and $y \leqslant z$, meaning y is a prefix of z can both be defined.

The bounded quantifiers are

$$\exists \xi \leqslant \eta\colon \quad \varphi \overset{\mathrm{def}}{=} \exists \xi \quad (\xi \leqslant \eta \,\&\, \varphi) \quad \text{and} \quad \forall \xi \leqslant \eta\colon \quad \varphi \overset{\mathrm{def}}{=} \neg \exists \xi \leqslant \eta\colon \quad \neg \varphi.$$

A formula $\mathrm{Tree}(\alpha) \overset{\mathrm{def}}{=} \forall y \leqslant \alpha\, \forall z \leqslant \alpha\, (\alpha z \,\&\, y \leqslant z \Rightarrow \alpha y)$ saying that α is a tree, is defined by quantifiers bounded by α. Note that according to this definition, the empty set is a tree. For technical reasons we shall also introduce tree variables $\Delta, \Delta_1, \ldots$ ranging over finite trees, defining $\exists \Delta\colon \varphi \overset{\mathrm{def}}{=} \exists \alpha\ [\mathrm{Tree}(\alpha) \,\&\, \varphi]$, $\forall \Delta\colon \varphi \overset{\mathrm{def}}{=} \neg \exists \Delta\, \neg \varphi$.

Theorem 4.1. *Each formula $\varphi = \varphi(y, \beta)$, where y and β are some vectors of variables, can be effectively represented as*

$$\varphi \equiv \exists \Delta_1 \forall \Delta_2 \ldots \exists(\forall)\Delta_m\colon \quad \varphi_0(y, \beta, \Delta_1, \ldots, \Delta_m), \quad m \geqslant 0 \tag{6}$$

where φ_0 contains only bounded quantifiers and $\Delta_1, \ldots, \Delta_m$ are tree variables.

Proof. The following is proved in [4, Lemma 2] and is useful here.

Lemma 4.2. *For each $Q1, Q2 \in \{\exists, \forall\}$ we have*

$$Q1\alpha \leqslant \beta\, Q2\gamma\colon \psi(\alpha, \beta, \gamma) \;=\; Q2\gamma'\, Q1\alpha \leqslant \beta\, Q2\gamma \leqslant \gamma'\colon \psi'(\alpha, \beta, \gamma, \gamma')$$

where

$$\psi'(\alpha, \beta, \gamma, \gamma') = \begin{cases} \psi(\alpha, \beta, \gamma') \;\&\; \forall x \leqslant \gamma'\,(x \in \gamma \Rightarrow x \in \gamma') & \text{for } Q2 = \exists, \\ \forall x \leqslant \gamma'\,(x \in \gamma \Rightarrow x \in \gamma') \Rightarrow \psi(\alpha, \beta, \gamma') & \text{for } Q2 = \forall. \end{cases}$$

This lemma is in fact a determinancy theorem for a game on a very special kind of tree.

Now let us reduce φ to $\exists\alpha 1\, \forall\alpha 2 \ldots \exists(\forall)\alpha m\colon \varphi' = \exists\alpha 1\colon \psi$ with φ' bounded. Then we substitute $\exists\alpha 1\colon \psi$ as $\exists\Delta_1\, \exists\alpha 1 \leqslant \Delta_1\colon \mathrm{Tree}(\Delta_1)\ \&\ \psi$. The remaining part of the proof is by induction on m, since according to our lemma the bounded quantifier $\exists\alpha 1 \leqslant \Delta_1\colon$ and the bounded quantifiers appearing in the definition of $\mathrm{Tree}(\Delta_1)$ can be successively shifted by the unbounded quantifiers $\forall\alpha 2, \exists\alpha 3, \ldots$. □

When defining for $n \geqslant 0$, $\mathbb{B}_0 = \exists\mathbb{B}_0 = \forall\mathbb{B}_0 =$ formulas with bounded quantifiers only; $\varphi \in \exists\mathbb{B}_{n+1}$ iff $\varphi \equiv \exists\alpha\colon \psi$ for some variable α and some $\psi \in \forall\mathbb{B}_n$; $\varphi \in \forall\mathbb{B}_n$ iff $\neg\varphi \in \exists\mathbb{B}_n$; we obtain an infinite hierarchy (see [13]). A φ satisfying (6) belongs to $\exists\mathbb{B}_n$. Then defining

$$\mathbb{B}_{n+1} = (\exists\mathbb{B}_{n+1} \cup \forall\mathbb{B}_{n+1}) \setminus (\exists\mathbb{B}_n \cup \forall\mathbb{B}_n)$$

we have $\varphi \in \mathbb{B}_n$ if and only if a minimal number of unbounded quantifiers in a normal form of φ equals n.

In the sequel we shall suppose that $\Sigma \subseteq \{\mathrm{false}, \mathrm{true}\}^v$. Then a Σ-tree t defines the predicates $P_1, \ldots, P_v$ on K^* by the equality $t(y) = P_1 y \times \cdots \times P_v y$ for $y \in K^*$. Conversely the predicates define a Σ-tree. We shall denote $\varphi(P_1, \ldots, P_v) = \mathrm{true}$ as $t \vdash \varphi$. For an automaton M by φ_M we shall understand such a formula, that $t \vdash \varphi$ iff $t \vdash M$. For a weak Muller a.f.a. M the φ_M is a weak formula, and each weak formula $\varphi = \varphi_M$ for some weak Muller a.f.a. (see [5]).

5. Representation theorems

First we shall prove the following.

Theorem 5.1. *For each $m = 0, 1, 2, \ldots$ and each Rabin a.f.a. M of index m, the formula $\varphi_M \in \mathbb{B}_m$ holds, i.e. φ_M has at most m unbounded quantifiers.*

Proof. Case $m = 0$. Then $\varphi_M = \text{false}$ or $\varphi_M = \text{true}$ so it has at most 0 unbounded quantifiers.

Case $m > 0$. One can suppose that M is structured for, e.g. $\emptyset = L_0 \subset L_1 \subset L_2 \subset \cdots \subset L_m \subset L_{m+1} = Q$ and $M = \langle L(K \times Q), \Sigma, \delta, ?, \Omega \rangle$ has the initial state unspecified.

For a $u \in P(Q)$ we shall construct a formula $\varphi_{M,u}[y]$ depending on a parameter $y \in K^*$, such that $t \vdash \varphi_{M,u}[y]$ iff for each $q \in u$ and each $t_y = t|\{z \geqslant y\}$: $t_y \vdash M_q$. Here M_q is the M with the initial state specified as q.

(A) Case when $D = (L_{m+1}, L_m) \notin \Omega$. Choose some coding of the elements of $P(Q)$, e.g. the coding is contained in $\{\text{false}, \text{true}\}^e$ and set $M'' = \langle L(K \times Q), \Sigma, \delta, ?, \Omega'' \rangle$, where $\Omega'' = \Omega \cup \{D\}$. To be more specific, M'' is structured for

$$\emptyset = L_0 \subset L_1 \subset L_2 \subset \cdots \subset L_{m-1} \subset L_{m+1} = Q$$

since the pairs $(L_{m+1}, L_m) = D$ and (L_m, L_{m-1}) are converted to one (L_{m+1}, L_{m-1}). $\text{Index}(M'') = m - 1$.

Now we can see that for $w \subseteq L_m$ we have $\varphi_{M'',w}[z] = \varphi_{M,w}[z]$. Hence the desired formula is

$$\varphi_{M,u}[y] = \exists \Delta \{\cdots\}$$

where $\{\cdots\}$ is $\{y \in \Delta$ and $\text{Tree}(\Delta)$ and for some run r of SM on Δ_y, starting in y with u, we have for each state w and each z in the frontier of Δ_y: the $w = r(z)$ follows $w \subseteq L_m$ and $\varphi_{M'',w}[z]\}$.

Here $\Delta_y = \{v\colon v \in \Delta \ \& \ y \leqslant v\}$ is a finite tree defined by a given value of the variable Δ (see Fig. 1).

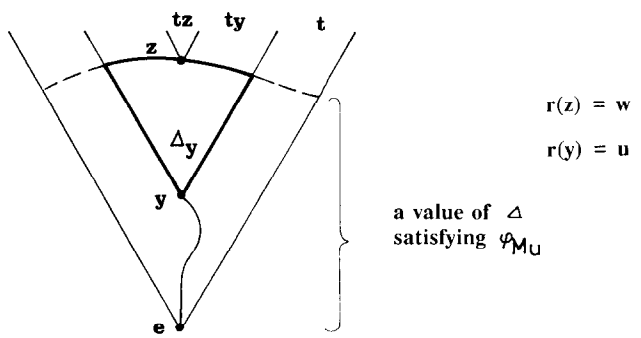

Fig. 1.

By Lemma 3.2 if $t \vdash \varphi_{M,u}[y]$ then t satisfies the formula. The converse: for proof there remains to code the run r by a well known procedure (see [9] or [10]) and express $\{\cdots\}$, depending on $\varphi_{M'',w}[z]$, by quantifiers bounded by Δ.

Now $\varphi_M = \varphi_{M,q0}[e]$ where e is the root of K^*.

(B) Case when $\{D\} \subseteq \Omega$. Construct φ_M, for the complementary automaton M'. Then $\varphi_M = \neg \varphi_{M'}$. According to the induction in both cases $\varphi_M \in \mathbb{B}_M$. □

Now we shall prove the converse.

Theorem 5.2. *Let $\varphi = \varphi[P_1, \ldots, P_v]$ be a weak formula without individual, finite set and tree variables occurring freely. For each $m = 0, 1, 2, \ldots$, if $\varphi \in \mathbb{B}_m$ then $\varphi = \varphi_M$ for some weak Rabin a.f.a. of index m.*

Proof. If $m = 0$ then $\varphi \in \mathbb{B}_0$ is a bounded formula. Then it is quantifier-free, since no variables exists to bound the quantified variables. Hence φ is a Boolean combination of the predicates. Since no individual variable exists freely in the predicates, they must be "false" or "true". Hence $\varphi \equiv$ false or $\varphi \equiv$ true. Then for an a.f.a. M of index 0, accepting none or accepting all trees respectively, we have $\varphi = \varphi_M$.

Now suppose $m > 0$. Convert φ to a form (6). Suppose

$$\varphi = \varphi[P_1, \ldots, P_v] \equiv \exists(\forall)\Delta_1 \, \forall\Delta_2 \ldots \exists\Delta_m : \quad \varphi_0(\Delta_1, \ldots, \Delta_m).$$

If the last quantifier is $\forall\Delta_m$ then take $\neg\varphi$.

Let $\Delta'_1, \ldots, \Delta'_m$ be values of $\Delta_1, \ldots, \Delta_m$. Choose a new variable Δ and its value Δ' such that $\Delta'_1 \leqslant \Delta', \ldots, \Delta'_m \leqslant \Delta'$. Let us convert φ to a form (6). Let us suppose the last quantifier in $\exists(\forall)$ is $\exists$. If it is $\forall$ then use $\neg\varphi$. Then the formula $\exists\Delta_m \varphi_0$ is equivalent to $\exists\Delta \varphi'$ where

$$\varphi' \equiv \exists\Delta_1 \leqslant \Delta : \quad \varphi_0(\Delta_1, \ldots, \Delta_m) \,\&\, \Delta_1 \leqslant \Delta \,\&\, \cdots \,\&\, \Delta_{m-1} \leqslant \Delta. \tag{7}$$

We can represent $\exists\Delta_1 \leqslant \Delta : \varphi_0(\Delta_1, \ldots, \Delta_m)$ by a classical finite automaton A acting on the tree $t|\Delta'$ (for details see [11]). Using this automaton we may construct an a.f.a. acting as A on $t|\Delta'$, and above the frontier of Δ' repeating its states for ever and sending copies in all directions. Let us construct an a.f.a. M_2 testing $\Delta'_1 \leqslant \Delta' \,\&\, \cdots \,\&\, \Delta'_{m-1} \leqslant \Delta'$. When $\Delta'_1(x) =$ false and $\Delta'(x) =$ true then the automaton M_2 enters in an accepting state, repeats it for ever and sends copies in all directions.

The formula (7) is represented by a product of M_1 and M_2 with conditions given by a Rabin pair of type $(F, \emptyset)$.

The rest of the proof is by a successive use of Lemma 3.4 and the complementation theorem for a.f.a. □

The normal form (6) from Theorem 4.1 is a precise recipe for constructing the automaton. Following Büchi's way of thinking (see [1]) one can call this form "an automaton".

Note that the assumption that the formula φ contains no free variables is essential for Theorem 5.2. The formula Tree(α) is bounded but is, and must be, represented by an a.f.a. of index equal 2.

If a formula φ contains free variables then we can remove them in the following manner. First, for each individual variable y and for each finite set variable α we add a unary predicate P_y and P_α, respectively. Then the formula

$$\begin{aligned}\varphi'' &= \varphi''[P_1, \ldots, P_v, P_y, P_\alpha] \\ &= \exists y\, \exists \alpha \quad \{\varphi[P_1, \ldots, P_v](y, \alpha) \,\&\, \forall z \\ &\qquad \{P_y z \equiv (z = y) \,\&\, P_\alpha z \equiv \alpha z\}\}\end{aligned} \tag{8}$$

contains no free variables. The second part of the formula enforces

$$P_y = y \quad \text{and} \quad P_\alpha = \alpha. \tag{9}$$

Now let φ belong to $\mathbb{B}_m$ for some $m \geqslant 0$. Let us suppose that the first unbounded quantifier in form (6) of the φ is either $\exists$ or none (the last takes place for $m = 0$). For $\forall$ use $\neg\varphi$. Then the formula φ'' given by (8) belongs to $\mathbb{B}_{m''}$ where $m'' = \max(m, 2)$. That is because we have $\exists$ and $\forall$ placed consecutively one after the other. Hence by Theorem 5.2 the formula φ'' is represented by a weak Rabin a.f.a. M of index m''. According to (9) the same automaton represents φ. Hence we have the following.

Theorem 5.3. *For each formula $\varphi = \mathbb{B}_m$ where $m \geqslant 0$, containing at least one free weak variable there exists a weak Rabin a.f.a. M of index* $\max(m, 2)$ *such that $\varphi \equiv \varphi_M$.*

References

[1] J.R. Büchi, The monadic second order theory of ω_1, in: J.R. Büchi and D. Siefkes, eds., *The Monadic Second Order Theory of All Countable Ordinals, Decidable Theories II*, Lecture Notes in Mathematics **328** (Springer, Berlin, 1973) 1-27.

[2] Y. Gurewith and L. Harrington, Trees automata and games, in: *14th Symp. on Theory of Computation* (Association for Computing Machinery, 1982) 60-65.

[3] A.W. Mostowski, Regular expressions for infinite trees and a standard form of automata, in: *Computation Theory, Proc. 5th Symp.*, Zaborów, Poland, Lecture Notes in Computer Science **208** (Springer, Berlin, 1984) 157-168.

[4] A.W. Mostowski, Hierarchies of weak monadic formulas for two successor arithmetics, *J. Inform. Process. Cybern.* **23** (1987) 509-515.

[5] D.E. Muller, A. Saoudi and P.E. Schupp, Alternating automata and a weak monadic theory of the tree and its complexity, in: *Proc. ICALP 86*, Lecture Notes in Computer Science (Springer, Berlin, 1986) 275-283.

[6] D.E. Muller, P.E. Schupp, Alternating automata on infinite objects, determinancy and Rabins theorem (abstract), in: *Automata on Infinite Words*, Lecture Notes in Computer Science **192** (Springer, Berlin) 100-107.

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

[8] D. Niwiński, On fixed point clones, in: L. Kott, ed., *Proc. 13th ICALP* Lecture Notes in Computer Science **226** (Springer, Berlin) 465-473.

[9] M.O. Rabin, Decidability and definability in second-order theories, *Congress Intern. Math.* **1** (1970) 239-244.

[10] M.O. Rabin, Weakly definable relations and special automata, in: Y. Bar Hillel, ed., *Mathematical Logic and Foundations of Set Theory* (North-Holland, Amsterdam, 1972) 1-23.

[11] J.W. Thatcher and J.B. Wright, Generalised finite automata theory with an application to a second order logic, *Math. Systems Theory* **2**, 57-81.

[12] W. Thomas, A hierarchy of sets of infinite trees, in: *Theoretical Computer Sciences Proc.*, Lecture Notes in Computer Science **145** (Springer, Berlin, 1983) 335-342.