Theoretical Computer Sciences 54 (1987) 267-276
North-Holland

ALTERNATING AUTOMATA ON INFINITE TREES

David E. MULLER,

Department of Mathematics, University of Illinois, Urbana, IL 61801, U.S.A., and Department of Computer Science, University of Texas, Austin, TX 78712, U.S.A.

Paul E. SCHUPP *

Department of Mathematics, University, of Illinois, Urbana, IL 61801, U.S.A., and LITP, Université Paris 7, Paris Cedex, 05, France

Communicated by D. Perrin
Received January 1987

1. Introduction

The concept of an alternating automaton described in the fundamental article of Chandra, Kozen and Stockmeyer [4] generalizes the notion of nondeterminism by allowing states to be existential or universal and has since been utilized by several authors. In this article we present a somewhat different conception of alternating automata and argue that this conception gives the 'natural' theory of automata on trees—whether finite or infinite. In the usual theory of nondeterministic automata the transition function is a one-to-many mapping from states to states. In our theory, the transition function is a mapping from the state set into the free distributive lattice generated by all the possible pairs (direction, state). Availability of the lattice operations makes complementation easy: Given an automaton M which is alternating in our sense, the dual automaton $\tilde{M}$ is obtained by simply dualizing the transition function and complementing the acceptance condition. It turns out that $\tilde{M}$ always accepts the complement of the language accepted by M—even on infinite trees. Alternating automata may be viewed as a sort of completion of nondeterministic automata. The interaction of state and direction is made clear by the lattice formalism and the free distributive lattice provides the smallest framework in which one can always dualize. Although they are a generalization of nondeterministic automata, alternating automata are more like deterministic automata in that complementation is easy. This article is the full version of the abstract in [14].

Although, as explained below, we believe that the main interest of our theory is for finite-state automata having 'weak' acceptance conditions, the concept of alternation gives a general 'formal' solution for the complementation problem. As clearly explained by Chandra, Kozen and Stockmeyer, the concept of an 'alternating

* This author's work was supported by NSF Grant DMS 83-02550.

0304-3975/87/$3.50

machine' applies to all the types of machines which one usually studies: finite-state automata, pushdown automata, Turing machines, etc. We treat these cases uniformly by considering automata with countable state sets. For example, the set of 'total states' of a pushdown automata is the set of all possible pairs consisting of the control state and the word written on the stack and the transition function operates on these pairs in the usual way. In general, we need not assume that the state set is finite nor that the transition function is even effectively calculable.

The proof that the dual automaton accepts the complementary language uses the existence of a winning strategy in a certain infinite game of perfect information. The fundamental theorm of Martin [9, 10] concerning such games states that one of the players always has a winning strategy provided only that the winning condition is Borel. Thus, the complementation result is also independent of the precise nature of the acceptance condition $\mathscr{F}$ and requires only that $\mathscr{F}$ is a Borel subset of Q^ω.

We next discuss what we do *not* do. The realization of a deep connection between complementation and the determinacy of games is found in Büchi [2]. One associates an infinite game with an automaton M working on an input tree t where one of the players plays for acceptance by M while the other plays for rejection. Büchi's ideas was that if one could prove that one of the players always has a winning strategy with 'bounded memory', then this strategy could itself be carried out by a finite-state automaton and the existence of an automaton accepting the complement would be shown. The existence of such a winning strategy has recently been proved by Gurevitch and Harrington [6], and independently by Muchnik [13]. The Gurevitch–Harrington theorem is already seen to be an indispensible tool in attempting to understand the full monadic theory of the tree and we wholeheartedly endorse this point of view.

We have stated above that completion is easy for alternating automata and that the proof of this result does not require a special winning strategy. This is indeed true but there is a well-known metaprinciple which, in the succinct formulation of Barry Commoner, states that "There is no free lunch". The essential conflict between the difficulties of complementation and of quantification persists in the case of alternating automaton at the level of the full monadic theory. Even when the state set is finite, the alternating automata which we define are 'infinite' machines in the sense that there are more and more copies of the automaton as the calculation proceeds. It is thus far from evident that a finite state alternating automaton can be simulated by a nondeterministic Rabin automaton. Our original motivation was to give a simpler proof of the fundamental theorem of Rabin [17] on the decidability of the full monadic theory of the binary tree. However, the best way to show that alternating automata can be simulated by Rabin automata is to use the Gurevitch–Harrington theorem and we consider this application as an illustration of the power of their method. This is fully worked out in the case of automata on infinite words in the article of Lindsay [8] and these ideas easily extend to the tree case.

Nonetheless, we do feel that alternating automata provide the 'natural' theory of automata on trees. As stated above, complementation is always easy for alternating automata. It is for quantification that one must pay in terms of increased complexity. One often does not want to consider the full monadic theory, which permits quantification over arbitrary sets, but the weak monadic theory (quantification only over finite sets), the first-order theory (quantification over individuals) or temporal logic, which has a very restricted quantification. Indeed, one descends in this hierarchy exactly by restricting quantification. Since Boolean operations are always easy for alternating automata, they become increasingly much more advantageous in a situation where quantification is restricted.

In joint work with Saoudi [15], we define a 'weak acceptance condition' and show that a collection of trees is definable in the weak monadic theory if and only if it is accepted by a finite-state alternating automaton using weak acceptance. This result both utilizes and simplifies the basic result of Rabin [18] characterizing the weak monadic theory of the tree. In that article we show that if a weak alternating automata accepts an input, it has a 'zero memory' accepting strategy. Alternating automata with weak acceptance also give a trivial proof of the exponential decidability of many temporal and dynamic logics [16].

This article is divided into two sections. In the first section we shall introduce the basic definitions for alternating automata on trees. In the second section we shall apply Martin's theorem to establish the complementation result.

This is one of a series of papers on alternating automata on trees. Our work has benefited in an essential way from discussions with many colleagues, including Leo Harrington, Ward Henson, Michael Makkai, André Muchnik, Dominique Perrin, Michel Parigot, Ahmed Saoudi and Alexii Simenov, and we would not have understood alternating automata without them. We are indebted to Leo Harrington and Michel Parigot for numerous conversations concerning the Gurevitch–Harrington theorem and to Alexii Simenov and André Muchnik for the conversations in which we learned about Muchnik's induction technique. During the writing of this paper we have benefited enormously from conversations with Ward Henson and Michel Parigot and their influence is felt throughout.

During the writing of this paper we became aware of the articles of Brzozowski and Leiss [3] on Boolean automata on finite words in which they take the state set to be a Boolean algebra—a point of view very similar to our own. Both finite state automata and Turing machines which are alternating is the sense of Chandra, Kozen and Stockmeyer have been considered on infinite words. See Lindsay [7, 8] and Miyano and Hayashi [10].

2. Alternating automata on trees

We consider the theory of alternating automata on the infinite k-ary tree, $k \geqslant 1$. In Rabin's theory [17] of nondeterministic finite-state automata on the binary tree, a single copy of the automaton begins in its initial state at the root of the tree. The automaton then splits into two copies, one moving to the left successor and the other moving to the right successor. The states of the two copies are given by a nondeterministic choice from the pairs of possibilities allowed by the transition function. In Rabin's notation, if the automaton is in state q_0 reading the letter a, the value of the transition function for (q_0, a) might be $\{(q_1, q_2), (q_0, q_3)\}$ where the left (right) member of a pair denotes the next state of the automaton moving to the left (right) successor vertex. We can represent this situation in the lattice formulation by using the lattice $\mathscr{L}(\{0,1\} \times Q)$ generated by all the possible pairs (direction, state). Namely, we write

$$\delta_a(q_0) = (0, q_1) \wedge (1, q_2) \vee (0, q_0) \wedge (1, q_3)$$

(where, as usual, $\wedge$ has precedence over $\wedge$).

We interpret this expression as saying that the automaton has the choice of splitting into one copy in state q_1 going to the left successor *and* one copy in state q_2 going to the right successor, *or* of splitting into one copy in state q_0 going to the left *and* one copy in state q_3 going to the right. We note that both 'and' and 'or' are present in the conception of an automaton on the binary tree.

In the general case of an alternating automaton we allow $\delta_a(q)$ to be an arbitrary element of the free distributive lattice $\mathscr{L}(\{0,1\} \times Q)$. For example, the dual of the expression above is

$$\tilde{\delta}_a(q_0) = (0, q_1) \wedge (0, q_0) \vee (0, q_1) \wedge (1, q_3) \vee (0, q_0) \wedge (1, q_2) \\ \vee (1, q_2) \wedge (1, q_3).$$

This expression illustrates that we do not require the automaton to send copies in all direction (although at least one copy must go in some direction) and that several copies may go in the same direction. One may think of an alternating automaton as a sort of completion of a nondeterministic automaton. It is only by going to $\mathscr{L}(\{0,1\} \times Q)$ that one can always calculate the dual of a given transition function.

Before giving precise definitions concerning alternating automata we review our conventions on the k-ary tree T_K viewed as a structure. The vertex set of T_K is the set K^* of all words on the *direction alphabet* $K = \{0, 1, \dots, k-1\}$ with the empty word being the origin of the tree. Given a vertex v and a letter $l \in K$, there is an *edge* e with *label* l from v to vl, and vl is the *l-successor* of v. The *level* $|v|$ of a vertex v is the length of v as a word and is thus also the graph distance from the origin to v. We think of the edges in T_K as being labelled and the vertices as being unlabelled. Given a finite alphabet Σ, a k-ary *Σ-tree* t is T_K together with a function $\lambda : T_K \to \Sigma$ assigning a letter of Σ to each vertex of T_K. We use (T_K, Σ) to denote the collection of all k-ary Σ-trees. For simplicity, we consider the case $k = 2$ of binary trees unless otherwise specified.

We also fix our notation concerning lattices. If S is a set, then $\mathscr{L}(S)$ will denote the free distributive lattice generated by S. The characteristic property of $\mathscr{L}(S)$, which is equivalent to freeness, is that if J is any distributive lattice and if $\delta : S \to J$ is any function, then δ has a unique extension to a homomorphism $\delta' : \mathscr{L}(S) \to J$. We shall drop the prime notation and simply write δ for the extension also.

A *term* C is a conjunction of generators of $\mathscr{L}(S)$ where no generator occurs more than once. A term C *subsumes* a term C' if every generator which occurs in C' occurs in C. We shall sometimes find it convenient to identify C with the subset of generators occurring in C. Each element $e \in L(S)$ has a unique representation in *disjunctive normal form*, $e = \bigvee_i C_i$, where each C_i is a term and no C_i subsumes a C_k with $k \neq i$.

If $e = \bigvee_i \bigwedge_j s_{i,j}$ is an element of $\mathscr{L}(S)$, the *dual* of e is the element $\tilde{e} = \bigwedge_i \bigvee_j s_{i,j}$ obtained by interchanging $\wedge$ and $\vee$. (The latter element is not, of course, in disjunctive normal form.) We always write $\tilde{e}$ for the dual of e. If one expands $\tilde{e}$ into disjunctive normal normal form, say $\tilde{e} = \bigvee_e D_l$, then one sees that the collection of D_l consists exactly of the minimal selection sets choosing one generator from each term C_i of e. If $\delta : \mathscr{L}(S) \to J$ is a homomorphism defined by $\delta(s_i) = h_i$, then the *dual homomorphism* $\tilde{\delta}$ is defined by $\tilde{\delta}(s_i) = \tilde{h}_i$. The key property of dual homomorphisms is that if $\delta : \mathscr{L}(S) \to J$ is a homomorphism and $e \in \mathscr{L}(S)$, then $\widetilde{\delta(e)} = \tilde{\delta}(\tilde{e})$.

Definition 2.1. An *alternating automaton* on k-ary Σ-trees is a tuple

$$M = \langle \mathscr{L}(K \times Q), \Sigma, \delta, q_0, \mathscr{F} \rangle,$$

where K is the finite set of *directions*, Q is a countable set of *states*, Σ is a finite *input alphabet*, $\delta : \Sigma \times Q \to \mathscr{L}(K \times Q)$ is the *transition function*, $q_0 \in Q$ is the *initial state*, and the *acceptance condition* $\mathscr{F}$ is a Borel subset of Q^ω.

We now discuss acceptance by alternating automata. The intuitive idea is quite clear. An automaton M accepts an input t if there is an infinite sequence of choices such that the histories of all the individual machines which exist accept according to the condition $\mathscr{F}$. There are two equivalent ways of formalizing this notion. The first is to think in terms of the 'physical model' and to define the complete computation tree $T(M, t)$ of M on t. The branches in $T(M, t)$ represent the different possibilities for the choices of M.

We consider an example before giving a formal definition.

Example 2.2. Consider the automaton $M = \langle \mathscr{L}(K \times Q), \{a\}, \delta, q_0, \mathscr{F} \rangle$, where $Q = \{q_0, q_1, q_2\}$ and δ is defined by

$$\delta_a(q_0) = (0, q_0) \wedge (1, q_2) \ \vee \ (0, q_1);$$

$$\delta_a(q_1) = (0, q_1) \wedge (0, q_2) \wedge (1, q_2);$$

$$\delta_a(q_2) = (0, q_2).$$

The first three levels of the computation tree are illustrated in Fig. 1. At each vertex we have given the list of all the histories labelling the vertex. After the initial state there are two possibilities. Either there is an automaton in state q_0 at 0 with 0-*history* $(q_0, 0, q_0)$ up to level one and an automaton in state q_2 at 1 with history $(q_0, 1, q_2)$ (this is the possibility represented by the leftmost vertex of level one in

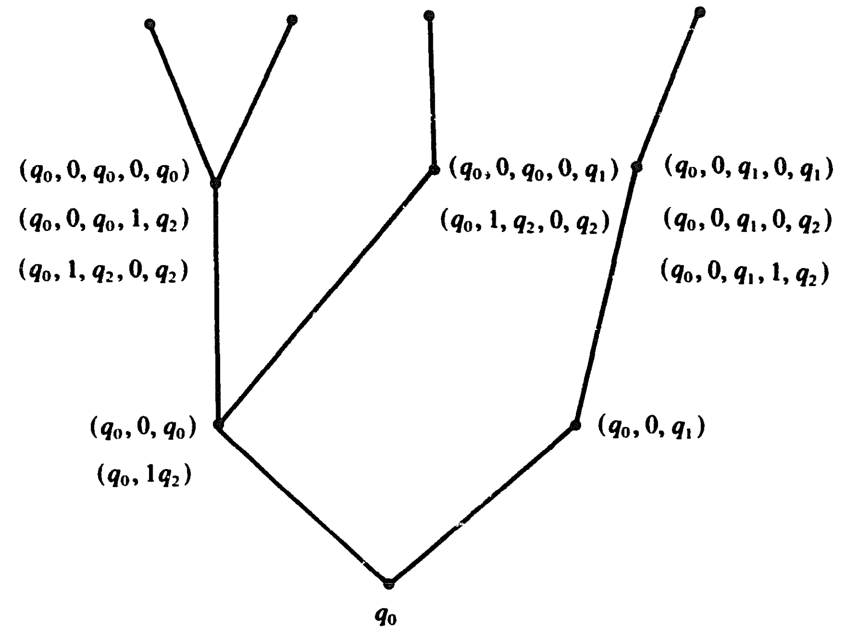

Fig. 1.

the computation tree) *or* there is a single automaton in state q_1 at 0 with history $(q_0, 0, q_1)$. Since on reading the letter a in state q_1 the automaton splits into three copies, on continuing the rightmost branch of the computation tree we would find three automata in existence, two at 00 and one at 01, with histories $(q_0, 0, q_1, 0, q_1)$, $(q_0, 0, q_1, 0, q_2)$ and $(q_0, 0, q_1, 1, q_2)$ respectively. We now give a precise definition using the lattice formalism.

Let t be a Σ-tree and let $M = \langle \mathscr{L}(K \times Q), \Sigma, \delta, q_0, \mathscr{F} \rangle$ be an alternating automaton with input alphabet Σ. We consider the transition function δ as a collection $\{\delta_a\}$ of transition functions where, for each $a \in \Sigma$, $\delta_a : Q \to \mathscr{L}(K \times Q)$. In the computation tree $T(M, t)$ of M on t, the vertices of level n in $T(M, t)$ will represent all the possibilities for choices of M up to level n in t. For each $n \geq 0$ we define the set of *n-histories* to be the set $H_n = q_0(K \times Q)^n$ of all strings consisting of q_0 followed by a string of length n from $K \times Q$. (An n-history will be the complete history of a single copy of the automaton up to level n, including the path taken by the copy.) If $h \in H_n$ and $g \in (K \times Q)$, let $hg \in H_{n+1}$ denote the concatenation of h and g. (Note that we will always write the lattice operation meet as $\wedge$ and that we use juxtaposition to denote concatenation of strings.) More generally, if $h \in H_n$ and $e \in \mathscr{L}(K \times Q)$, we denote by *he* the expression of $\mathscr{L}(H_{n+1})$ obtained by prefixing h to each generator of $\mathscr{L}(K \times Q)$ which occurs in e. If, for example, $h = (q_0, 0, q_1)$ and $e = (0, q_2) \wedge (1, q_3) \vee (0, q_4)$, then

$$he = (q_0, 0, q_1, 0, q_2) \wedge (q_0, q_1, 1, q_3) \vee (q_0, 0, q_1, 0, g_4).$$

If $h \in H_n$, say $h = y_0 k_1 y_1 \ldots k_n y_n$, where $y_0 = q_0$, each $y_i \in Q$, and each $k_i \in K$, then the *K-projection* of h is $\kappa(h) = k_1 \ldots k_n$ which thus defines a specific vertex v of the tree t. The *Q-projection* of h is $\rho(h) = y_0 \ldots y_n \in Q^{n+1}$ (giving the state history of a copy of the automaton which is present at v).

For each $n \geq 0$ we define a function $\delta_n : H_n \to \mathscr{L}(H_{n+1})$ as follows. Let a be the letter labelling the origin of t. Then, $\delta_1(q_0) = q_0\delta_a(q_1) \in \mathscr{L}(H_1)$. Now given an n-history h, there is only one way to continue it: find the letter a at the vertex v of t represented by h, find the state y_n of the copy of the automaton present at v which is represented by h, calculate the transition function $\delta_a(y_n)$, and prefix the result by h. In symbols, if $h = y_0k_1, \ldots, k_ny_n$ with $v = \kappa(h) = k_1 \ldots k_n$ and $a = \lambda(v)$, then $\delta_n(h) = h\delta_a(y_n)$ is in $\mathscr{L}(H_{n+1})$.

Definition 2.3. We define the computation tree $T(M, t)$ of M on t inductively as follows. The origin of $T(M, t)$ has label q_0. If $u \in T(M, t)$ is a vertex of level $n \geq 0$ already defined with label $\bigwedge_{i=1}^{m} h_i \in \mathscr{L}(H_n)$, then calculate

$$e = \bigwedge_{i=1}^{m} h_i\delta_{a_i}(h_i) \in \mathscr{L}(H_{n+1}).$$

Write e in disjunctive normal form as $e = \bigvee_{i=1}^{r} C_i$, where each C_i is a conjunction of generators of $\mathscr{L}(H_{n+1})$. Then u has r successor vertices, $u_1, \ldots, u_r$, and the label of the ith sucessor u_i of u is C_i. This completes the definition of $T(M, t)$. (The reader will see that this is exactly the computation which we made in Example 2.2.)

An *n-branch* β_n of $T(M, t)$ is a path of length n beginning at the origin of $T(M, t)$. A *branch* β is an infinite path. If u is the terminal vertex of an n-branch β_n, then u is labelled by a conjunction of n-histories, say $\bigwedge_{i=1}^{m} h_i \in \mathscr{L}(H_n)$. We say that each h_i *lies* along β_n. An *infinite history* is a sequence $h = (y_0, k_1, y_1, \ldots) \in q_0(K \times Q)^\omega$. Then *n-prefix* of h is $h_n = (y_0, k_1, y_1, \ldots, k_n, y_n)$. The infinite history h *lies along* the branch β if, for every $n \geq 0$, the n-prefix h_n of h lies along the n-branch β_n consisting of the first n edges of β. Each such history represents the history of some automaton in the physical interpretation.

We can now define acceptance by alternating automata. Let $M = \langle \mathscr{L}(K \times Q), \Sigma, \delta, q_0, \mathscr{F} \rangle$ be an alternating automaton and let $T(M, t)$ be the computation tree of M on t. An infinite history h is *accepting* if its Q-projection $\rho(h) \in \mathscr{F}$. A branch $\beta \in T(M, t)$ is *accepting* if every infinite history which lies along β is accepting. Finally, the automaton M accepts the input t if there exists an accepting branch β in $T(M, t)$. Intuitively, an accepting branch is exactly a sequence of choices of M such that all the machines arising in the sequence have accepting histories. (One branch of the computation tree takes into account the choices of M at all vertices of the input tree.) As usual, the *language* $L(M)$ *accepted by* M is the set of all Σ-trees accepted by M.

3. Games, duality and complementation

In this section we shall give the game-theoretic interpretation of acceptance and prove that if M is an alternating automaton, then the dual automaton $\tilde{M}$ accepts the complement of the language accepted by M. Given M and an input tree t we define a particular game $G(M, t)$ with players P and $\tilde{P}$. Intuitively, P plays for acceptance by M while $\tilde{P}$ plays for rejection. In the first move, P chooses a term from $\delta(a_0, q_0)$ where a_0 is the letter labelling the origin of t. (Formally, one can enumerate the set $\mathscr{C}$ of possible terms (conjunctions of generators) of $\mathscr{L}(K \times Q)$ and P plays a number from $\mathscr{C}$.) Then $\tilde{P}$ chooses a generator $k_1 y_1$ occurring in the term chosen by P. At move $(2n-1)$, the sequence $h_n = q_0 k_1 y_1 k_2 y_2 \ldots k_n y_n$ already chosen by $\tilde{P}$ is an n-history. Let $v_n = \kappa(h_n)$ be the vertex of t defined by h_n and let $a_n = \lambda(v_n)$ be its label. Then P chooses a term from $\delta(a_n, y_n)$ and $\tilde{P}$ subsequently chooses a generator occurring in the term chosen by P. If either player violates the rule restricting his choices, he immediately loses. Assuming that both players follow the rules, the infinite sequence of choices made by $\tilde{P}$ defines an infinite history h. Player P wins if the Q-projection $\rho(h) \in \mathscr{F}$. Otherwise, $\tilde{P}$ wins.

For a detailed discussion of such infinite games of perfect information see Moschovakis [12]. A game is said to be *determined* if one of the players has a winning strategy. The basic fact about such games is the deep theorem of Martin [9, 10] that games in which the winning condition is Borel are determined. If X is a countable set, one makes X^ω into a complete metric space by defining the distance between distinct sequences $r, s \in X^\omega$ by $d(r, s) = 1/(1+m)$, where m is the last natural number such that $r(m) \neq s(m)$. The game $G(M, t)$ which we have defined is Borel as long as $\mathscr{F}$ is a Borel subset of Q^ω.

A *strategy* β for the player P is a function from the set H of all finite histories to the set $\mathscr{C}$ of possible terms. Given β, the *list of possible situations* according to β consists of q_0, then, all generators occurring in $\beta(q_0)$, then the possible sequences $q_0 k_1 y_1 h_2 y_2$ where $q_0 k_1 y_1$ occurs in the list of possibilities of length one and $k_2 y_2$ occurs in $\beta(q_0 k_1 y_1)$, then the possibilities of length three, etc. We see that there is a one-to-one correspondence between strategies β for P in $G(M, t)$ and branches in the computation tree $T(M, t)$. Thus we have the following lemma.

Lemma 3.1. *The automaton M accepts an input t if and only if P has a winning strategy in the game $G(M, t)$.*

We define the dual automaton of M as follows.

Definition 3.2. Given an alternating automaton $M = \langle \mathscr{L}(K \times Q), \Sigma, \delta, q_0, \mathscr{F} \rangle$, the *dual automaton* is

$$\tilde{M} = \langle \mathscr{L}(K \times Q), \Sigma, \tilde{\delta}, q_0, \bar{\mathscr{F}} \rangle$$

where $\tilde{\delta}$ is the function obtained by dualizing δ and $\bar{\mathscr{F}} = Q^\omega - \mathscr{F}$ is the complement of $\mathscr{F}$.

Now a strategy $\tilde{\beta}$ for the player $\tilde{P}$ in the game $G(M, t)$ chooses an $(n+1)$-history lying along each $(n+1)$-branch of $T(M, t)$ in such a way that choices at level n are extended. Given an expression $\delta(a, q)$ in disjunctive normal form, the terms of $\tilde{\delta}(a, q)$ consist of the minimal choice sets which choose a generator from each term.

Thus, a strategy for $\tilde{P}$ in $G(M, t)$ gives a strategy for the first player in the game $G(\tilde{M}, t)$ defined by the dual automaton. We verify this as follows.

Definition 3.3. Let $T(M, t)$ be the computation tree of M on t. Let $n \geq 0$, and let $C_1, \ldots, C_m$ be the terms of $\mathscr{L}(H_n)$ which label the vertices of level n in $T(M, t)$. *The total expression of level n in $T(M, t)$* is $e_n = \bigvee_{i=1}^{m} C_i$.

Lemma 3.4. *Let $T(M, t)$ and $T(\tilde{M}, t)$ be the computation trees of M and $\tilde{M}$ on t. If e_n is the total expression of level n in $T(M, t)$, then the total expression of level n in $T(\tilde{M}, t)$ is $\tilde{e}_n$.*

Proof. The proof is by induction on n. The result holds for $n = 0$ since $e_0 = q_0 = \tilde{e}_0$. Assume that the result holds for a given value of n. Note that one calculates the total expression of level $(n+1)$ in $T(M, t)$ by applying δ_n to e_n, and in $T(\tilde{M}, t)$ one applies $\tilde{\delta}_n$ to the total expression of level n in $T(\tilde{M}, t)$ which, by the induction hypothesis, is $\tilde{e}_n$. Thus, $\tilde{\delta}_n(\tilde{e}_n) = \tilde{e}_{n+1}$. □

Lemma 3.5. *Let $T(M, t)$ and $T(\tilde{M}, t)$ be respectively the computation trees of M and $\tilde{M}$ on an input t. For every $n \geq 0$, the terms labelling the vertices of level n in $T(\tilde{M}, t)$ are exactly the minimal choice sets of n-histories for $T(M, t)$. If v is a vertex of level n in $T(\tilde{M}, t)$, say labelled by S, then the labels of the successors of v are exactly the minimal choice sets of $(n+1)$-histories which extend S.*

Proof. By the previous lemma, the total expression $e'_n = \bigvee_i S_j$ of level n in $T(\tilde{M}, t)$ is the dual of the total expression $e_n = \bigvee_i C_i$ of level n in $T(M, t)$. An easy induction shows that e_n and e'_n are in disjunctive normal form as written. Thus the S_j are exactly the minimal choice sets of generators (n-histories) belonging to the C_i. By the definition of the construction of a computation tree, a vertex of level $n+1$ labelled by a term S is the successor of a vertex labelled by the set of n-prefixes of elements in S. □

We note that the previous two lemmas immediately yield the relation between duality and complementation for automata working on finite k-ary trees where we consider alternating automata of the form $M = \langle \mathscr{L}(K \times Q), \Sigma, \delta, q_0, F \rangle$ and, as usual for automata on finite objects, $F \subseteq Q$ is a set of final states. In this case, the dual automaton is $\tilde{M} = \langle \mathscr{L}(K \times Q), \Sigma, \tilde{\delta}, q_0, \bar{F} \rangle$. We have the following corollary.

Corollary 3.6. *Let M be an alternating automaton on finite k-ary trees. Then the dual automaton $\tilde{M}$ accepts the complement of the language accepted by M.*

Lemma 3.7. *The automaton $\tilde{M}$ accepts an input tree t if and only if the player $\tilde{P}$ has a winning strategy for the game $G(M, t)$.*

Proof. As noted above, a strategy $\tilde{\beta}$ for $\tilde{P}$ chooses, for each n-branch β_n of $T(M, t)$, an n-history lying along β_n in such a way that previous choices are extended. For successive values of n one can select a minimal choice set among the histories chosen by β in such a way that the minimal choice set selected at stage $n-1$ is extended. Given that $\tilde{\beta}$ is winning for $\tilde{P}$, this defines an accepting branch of $T(\tilde{M}, t)$. Conversely, if one is given an accepting branch γ of $T(\tilde{M}, t)$, a winning strategy for $\tilde{P}$ is to select histories which lie along γ. $\square$

Thus, the only infinitary principle which is needed in our analysis is exactly the existence of a winning strategy for one of the two players in $G(M, t)$. This is provided by Martin's theorem and we have established our main result.

Complementation Theorem. *Let* $M = \langle \mathscr{L}(K \times Q), \Sigma, \delta, q_0, \mathscr{F} \rangle$ *be an alternating automaton on k-ary Σ-trees. Then the dual automaton* $\tilde{M} = \langle \mathscr{L}(K \times Q), \Sigma, \tilde{\delta}, q_0, \bar{\mathscr{F}} \rangle$ *accepts the complement of the language accepted by M.*

References

[1] J.R. Büchi, On a decision method in restricted second-order arithmetic, in: *Proc. 1960 Internat. Congress on Logic, Methodology and Philosophy of Science* (Stanford University Press, 1960) 1-11.

[2] J.R. Büchi, Using determinacy of games to eliminate quantifiers, in: Lecture Notes in Computer Science 56 (Springer, Berlin, 1977) 367-378.

[3] J. Brzozowski and E. Leiss, On equations for regular languages, finite automata, and sequential networks, *Theoret. Comput. Sci.* **10** (1980) 19-35.

[4] A. Chandra, D. Kozen and L. Stockmeyer, Alternation, *J. ACM* **28** (1981) 114-133.

[5] D. Gale and F.M. Stewart, Infinite games with perfect information, in: *Contributions to the Theory of Games*, Annals of Mathematical Studies **28** (Princeton University Press, 1953) 245-266.

[6] Y. Gurevitch and L. Harrington, Trees, automata and games, in: *Proc. ACM Symp. on the Theory of Computing*, San Francisco, CA (1982) 60-65.

[7] P. Lindsay, Alternation and ω-type Turing acceptors, *Theoret. Comput. Sci.*, **43** (1986) 107-115.

[8] P. Lindsay, On alternating ω-automata, *Theoret. Comput. Sci.*, to appear.

[9] D.A. Martin, Borel determinacy, *Ann. of Math.* **102** (1975) 363-371.

[10] D.A. Martin, A purely inductive proof of Borel determinacy, in: *Proc. Symp. in Pure Mathematics* **42** (American Mathematical Society, Providence, RI, 1985) 303-308.

[11] S. Miyano and T. Hayashi, Alternating automata on ω-words, *Theoret. Comput. Sci.* **32** (1984) 321-330.

[12] Y.N. Moschovakis, *Descriptive Set Theory* (North-Holland, Amsterdam, 1980).

[13] A.A. Muchnik, Games on infinite trees and automata with dead-end markers—a new proof of the decidability of the monadic theory of two successors, *Semiotika* (1985) 17-40 (in Russian).

[14] D.E. Muller and P.E. Schupp, Alternating automata on infinite objects, determinacy and Rabin's Theorem (abstract), in: *Automata on Infinite Words*, Lecture Notes in Computer Science **192** (Springer, Berlin, 1985) 100-107.

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

[16] D.E. Muller and P.E. Schupp, Alternating automata and the exponential decidability of an extended branching-time temporal logic, submitted.

[17] M.O. Rabin, Decidability of second-order theories and automata on infinite trees, *Trans. Amer. Math. Soc.* **141** (1969) 1-35.

[18] 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.