PART III

AUTOMATA ON INFINITE TREES

ALTERNATING AUTOMATA ON INFINITE OBJECTS, DETERMINACY AND RABIN'S THEOREM

David Muller, University of Illinois
Paul E. Schupp, University of Illinois, LITP, Université Paris VII

In this article we give an outline of a new theory of automata on infinite trees (or indeed, finite trees) which allows us to give a simple proof of Rabin's Theorem that the monadic theory of the binary tree is decidable. The concept of an alternating automaton described by Chandra, Kozen and Stockmeyer [1] generalizes the notion of nondeterminism by allowing states to be existential, universal or negating. We present a different notion of an alternating automaton in which the transition function is a homomorphism into a free distributive lattice. The availability of the lattice operations makes complimentation easy, even for infinite trees. Given an automaton M which is alternating in our sense, one can define the dual automaton M which always accepts the compliment of the language accepted by M. The proof of this fact directly uses the determinacy of certain infinite games and does not require an effective version of determinacy. It is not immediately obvious that the alternating automata which we define are equivalent to ordinary nondeterministic automata on infinite trees. We need this result in order to prove Rabin's theorem. To establish the result we prove a "uniformization theorem" which shows that certain nondeterministic automata have a "sufficiently uniform" strategy for acceptance. Gurevich and Harrington [2] have recently given a new proof of Rabin's theorem in which they do not reformulate the automata theory but instead prove that a special form of "forgetful determinacy" holds for a game associated directly with a nondeterministic automaton. The result which we prove is weaker in that the choices in the strategy may depend on an unbounded amont of information about the past history. This allows the strategy to be simply defined and verified.

One result of our method is that there is an effective construction which associates with each sentence ϕ of the monadic theory of the k-any tree a sentence ϕ' of the monadic theory of the natural numbers $\mathbb{N}$ such that ϕ is true if and only if ϕ' is true. A relativised version of this result gives a proper extension of the monadic theory of the binary tree which is still decidable. Finally, since the proof of the complimentation theorem depends only on the fact that the acceptance condition is Borel, the theory raises interesting questions concerning the possibility of using other acceptance conditions.

Our work has benefited greatly from many conversations with Ward Henson, André Muchnik and Alexi Simenov. Our proof of the uniformization theorem uses Muchnik's ingenous idea of decomposing the behaviour of a machine into the behaviours of the machines obtained by respectively converting each state into a dead-end marker and we learned this idea from Muchnik and Simenov. The influence of Ward Henson is felt throughout.

2. Alternating Automata on Strings and Trees

We define our notion of an alternating automaton and the definition of acceptance. Although it is the case of trees which really shows the naturalness of the lattice formulation, we begin by considering automata on finite strings. In a nondeterministic automaton the transition function, for each input letter, maps the present state to a collection of states. One member of this collection is thought of as being chosen nondeterministically. An input string is accepted if it is possible to make a sequence of nondeterministic choices which lead to a final state.

The dual possibility corresponds to "AND"–all choices must lead to acceptance. A convenient way to think of a "physical model" is to imagine that the automaton splits into several synchronized automata, one in each of the indicated states, and that all these machines must accept if the input sequence is to be accepted.

We want to combine both possibilities into a single transition of an automaton. We shall see shortly that this combination is, in fact, inherent in the idea of automata on trees. A nondeterministic automaton on finite strings is a tuple $M = \langle Q, \Sigma, \delta, q_0, F\rangle$ where Q is the state set and the transition function δ maps $Q \times \Sigma$ to $P(Q)$. In our sense, an alternating automaton on finite strings is a tuple

$$M = \langle L(Q), \Sigma, \delta, q_0, F\rangle$$

with transition function $\delta : Q \times \Sigma \to L(Q)$ where $L(Q)$ is the free dsitributive lattice generated by Q. Formally, $L(Q)$ has two binary operations : "meet", $\wedge$, and, "join", $\vee$, which are subject to all the usual laws. The easiest way to think of $L(Q)$ is to think of the elements of Q as propositional variables and to calculate as in Boolean algebra except that there is no negation in $L(Q)$.

For an alternating automaton on strings, if $a \in \Sigma$ is an input letter, the restriction δ_a of δ to $Q \times \{a\}$ is a function $\delta_a : Q \to L(Q)$. An : ordinary nondeterministic automaton corresponds to the special case where, for each pair $q \in Q$, $a \in \Sigma$, the function $\delta_a(q)$ is a disjunction of single states. If, more generally, $\delta_a(q) = \bigvee_i \bigwedge_j q_{i,j}$ we can interpret this expression by thinking that one of the indices i is chosen nondeterministically and that the automaton then splits into several copies, one for each state $q_{i,j}$. For example, if

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

we think that if a copy of the automaton is in state q_0 and reads the letter a, then it has a choice of going into state q_1 or splitting into two copies, one in state q_2 and the other in state q_3.

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

A term C is a conjunction of generators of $L(S)$ where no generator occurs more than once. Each element $e \in L(S)$ has a unique representation in disjunctive normal (up to the order of terms) form, $e = \vee C_l$, where no term C_l subsumes a C_k with $k \neq l$. We shall write $e = \bigvee_i C_i$ or $e = \bigvee_i \bigwedge_j s_{i,j}$ without indicating that the range of j depends on i.

If $e = \bigvee_i \bigwedge_j s_{i,j} \in L(S)$, the dual of e is the element $\tilde{e} = \bigvee_i \bigwedge_j s_{i,j}$ obtained by interchanging $\wedge$ and $\vee$. We always write $\tilde{e}$ for the dual of e. If $\delta : L(S) \to H$ is a homomorphism defined by $\delta(a_i) = h$ then the dual homomorphism $\tilde{\delta}$ is defined by $\tilde{\delta}(s_i) = \widetilde{h_i}$. It is easy to prove that $\tilde{\delta}(\tilde{e}) = \widetilde{\delta(e)}$.

We now consider the theory of alternating automata on the infinite k-ary tree, $k \geq 1$. For convenience we assume that $k = 2$. In Rabin's theory [4], a nondeterministic automaton on the infinite binary tree is a tuple

$$M = \langle Q, \Sigma, \delta, q_0, F \rangle$$

where, for each letter $a \in \Sigma$, the transition function $\delta_a: Q \to P(Q \times Q)$.

A single copy of the automaton begins at the root of the tree in its initial state q_0. The automaton then splits into two copies, one moving to the left successor vertex and the other moving to the right successor vertex according to the possibilities of the transition function. If for example the automaton starts in q_0 and reads the letter a, one might have $\delta_a(q_0) = \{(q_1, q_0), (q_3, q_4)\}$ where the left (right) member of a pair denotes the next state of the automaton moving to the left (right) successor. We can represent this situation in the lattice formulation by using the lattice $L(\{0,1\} \times Q)$ generated by all the possible pairs (direction, state). Namely, we write

$$\delta_a(q_0) = (0, q_1) \wedge (1, q_0) \vee (0, q_3) \wedge (1, q_4)$$

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

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 vertex <u>and</u> one copy in state q_2 going to the right successor vertex <u>or</u> of splitting into one copy in state q_0 going to the left <u>and</u> one copy in state q_3 going to the right. We thus note that both "or" and "and" are present in the idea of an automaton working on the binary tree.

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

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

This example illustrates that several copies may go in the same direction and that we do not require the automaton to send copies in all directions (although at least one copy must go in some direction). An alternating automaton is a type of completion of a nondeterministic automaton - it is only by going to $L(\{0,1\} \times Q)$ that one can always calculate the dual of a given expression.

<u>Definition</u> 2.1 - An <u>alternating automaton</u> on k-ary Σ-trees (the vertices are labelled from Σ) is a tuple

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

where $K = \{0, \ldots, k-1\}$ is the set of <u>directions</u>, Q is the set of states, Σ is the <u>input alphabet</u>, $\delta : Q \times \Sigma \to L(K \times Q)$ is the <u>transition function</u> and $F \subseteq P(Q)$ is the <u>family of accepting sets</u>.

Given an alternating automaton M and a Σ-tree t we want to define the computation tree $T(M,t)$ of M on t. Intuitively, the branches in $T(M,t)$ will represent the different possibilities for the choices of M and a vertex of level n in $T(M,t)$ will carry all the histories of the automata running at level n in t corresponding to the choice represented by the vertex. We consider an example

Example 2.1 Consider the automaton

$M = \langle L(K \times Q),\ \{a\},\ \delta,\ q_0,\ F\rangle$ 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 Figure 2.1. At each vertex we have given the list (conjunction) 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 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 the computation trees) or there is a single automaton in state q_1 at 1 with history $(q_0,\ 1,\ q_1)$.

For each $n \geqslant 0$ 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 is the complete history of a single copy of the automaton up to level n : including the path taken by the copy up to a vertex v of level n). Given an n-history η there is only one way to continue it : find the letter a at the vertex v of t represented by η, find the state y_n of the copy of the automaton present at v which is represented by η, calculate the transition function $\delta_a(y_n)$ and prefix the result by η.

Definition 2.2 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 $n \geqslant 0$ already defined with label

$\bigwedge_{i=1}^{m} \eta_i \in L(H_n)$, calculate $e = \bigwedge_{i=1}^{m} \eta_i\ \delta_a\ (\eta_i) \in L(H_{n+1})$.

Write e in disjunctive normal form as $e = \bigvee_{i=1}^{r} C_i$ where each C_i is a conjunction of generators ($(n+1)$-histories) of $L(H_{n+1})$. Then u has r successor vertices, $u_1,\ldots,u_r$ and the label of the i-th successor u_i of u is C_i. (This is exactly the computation which we made in Figure 2.1).

We can now define acceptance by an alternating automaton. An n-branch β_n of $T(M,t)$ is a reduced path beginning at the origin of $T(M,t)$. A branch β is an infinite reduced path. If u is the terminal vertex of an n-branch β_n then u is labelled by a conjunction of n-histories ; say $\bigwedge_{i=1}^{m} \eta_i \in L(H_n)$. We say that each η_i lies along β_n. An infinite history is a sequence $\gamma = (y_0,\ k_1,\ y_1,\ldots) \in q(K \times Q)^\omega$. The n-prefix of γ is $\gamma_n = (y_0,\ k_1,\ldots,k_n,\ y_n)$ where each $y_i \in Q,\ k_i \in K$. The infinite history γ lies along the branch β if for every $n \geqslant 0$, the n-prefix γ_n of γ lies along the n-branch β_n consisting of the first n edges of β. Given a branch β there may be many infinite histories which lie along β. Each such history represents the history of some automaton in the physical interpretation.

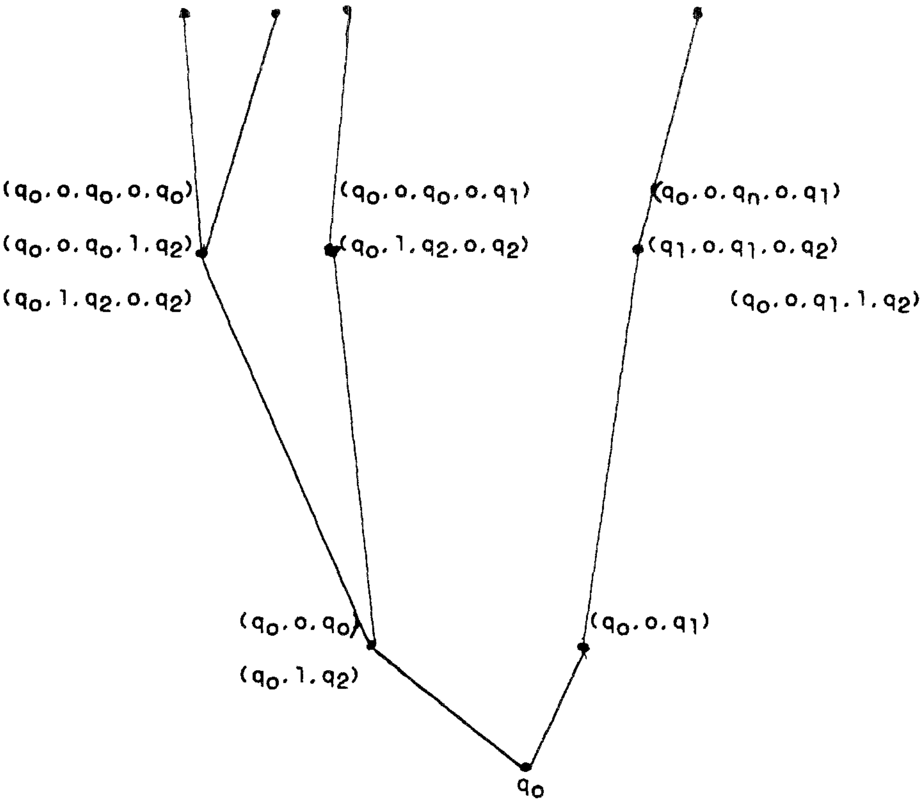

Figure 2.1

<u>Definition 2.3</u> If γ is an infinite history then $\mathrm{Inf}(\gamma)$ is the set of states which occur infinitely often in γ. A history γ is <u>accepting</u> if $\mathrm{Inf}(\gamma) \in F$ where F is the family of accepting sets. A branch β of $T(M,t)$ is <u>accepting</u> if every infinite history γ which lies along β is accepting. Finally, the alternating automaton M accepts 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. As usual, the language accepted by M is the set of all Σ-trees accepted by M.

It is necessary to verify that the class of languages accepted by alternating automaton is closed under the operations of union, complimentation, and projection. As usual, union is trivial. It turns out that complimentation is easy for alternating automata.

Definition 2.4 Given an alternating automaton $M = \langle L(K \times Q), \Sigma, \delta, q_0, F\rangle$ the dual automaton of M is $\widetilde{M} = \langle L(K \times Q), \Sigma, \widetilde{\delta}, q_0, \widetilde{F}\rangle$ where $\widetilde{\delta}$ is the function dual to δ and $\widetilde{F} = P(Q) - F$. Note that M is the dual of $\widetilde{M}$.

Theorem I (Complimentation Theorem)

Let $M = \langle L(K \times Q), \Sigma, \delta, q_0, F\rangle$ be an alternating automaton and let $\widetilde{M} = \langle L(K \times Q), \Sigma, \widetilde{\delta}, q_0, \widetilde{F}\rangle$ be the dual automaton. Then $\widetilde{M}$ accepts the compliment of the language accepted by M.

The proof of the complimentation theorem uses some simple lemmas concerning computation trees and the determinacy of certain infinite games of perfect information.

Let X be a finite set or the set $\mathbb{N}$ of natural numbers. The set X^ω of all infinite sequences of elements of X is a complete separable metric space under the metric defined by

$$d(\alpha, \beta) = \frac{1}{1 + \text{the least } n \text{ such that } \alpha(n) \neq \beta(n)}$$

We consider infinite games of the following sort. Let $Y \subseteq X^\omega$ be given. There are two players P_1 and P_2. Player P_1 plays first and chooses $x_1 \in X$. Then player P_2 chooses $x_2 \in X$. In general, on old numbered moves $(2i-1)$ player P_1 chooses $x_{2i-1} \in X$ while an even numbered moves player P_2 chooses $X_{2i} \in X$. The infinite sequence of choices defines $\sigma = (x_1, x_2, \ldots) \in X^\omega$. Player P_1 wins if $\sigma \in Y$ and P_2 wins otherwise. Using the axiom of choice one can construct a set Y such that neither player has a winning strategy.

A remarkable theorem of Martin [3] states that if Y is a Borel set then one of the players has a winning strategy.

It is easy to show that if M accepts a Σ-tree t then $\widetilde{M}$ rejects t. What requires proof is that, given an arbitrary Σ-tree t, one of M or $\widetilde{M}$ actually accepts t. We define two games, G and $\widetilde{G}$, which are played respectively on the computation trees $T(M,t)$ and $T(\widetilde{M},t)$. (Player P plays for acceptance by M while $\widetilde{P}$ plays for acceptance by $\widetilde{M}$). The player which has the winning strategy for G determines which of M or $\widetilde{M}$ accepts t. Here we use Martin's theorem directly, an effective version is not required.

3. Uniformization, Projection and Emptiness Algorithm

It is not immediately obvious that alternating automata are equivalent to nondeterministic automata (in the sense of Rabin) since the number of machines which may be present at a vertex is not bounded. We need to show that there is an effective construction which, when given an alternating automaton M constructs a nondeterministic automaton which is equivalent to M. Indeed, this is the only method we have to demonstrate closure under projection. Also, the result allows us to give a very simple proof that the emptiness problem is decidable for alternating automata.

We consider the relationship between automata working on k-any trees with direction set $K = \{0,\ldots,k-1\}$ and nondeterministic automata working on AxK-trees with more branching. (We now write the set of directions before the word "tree".) Given a vertex v in an AxK-tree t, the K-projection $\rho(v)$ is obtained by erasing the A-component of each letter in v to obtain a word $v' \in K^*$. (Thus $\rho((a,k_2)(a_2,k_2)) = (k_1,k_2)$.

Definition 3.1 An AxK-tree t labelled from an alphabet Σ is K-uniform if all the vertices of t which have the same K-projection are labelled by the same letter from Σ.

A K-uniform AxK-tree t labelled from Σ thus determines a unique K-tree t' and conversely. Given an alternating automaton M working on K-trees, it is easy to "unfold" M to obtain, for an appropriate set A a nondeterministic automaton N on AxK-trees which is equivalent to M in the following sense : for every K-uniform AxK-tree labelled from Σ, N accepts t if and only if M accepts the K-projection $t' = \rho(t)$.

The problem thus becomes the following : given a nondeterministic automaton N on AxK-trees one wants to construct a nondeterministic automaton P on K-trees which is equivalent to N in the sense above. In order to carry out such a construction, one must consider more information than just the K-projection. Let R be a regular equivalence relation on the set H^* of finite histories (that is, there exists a finite automaton E on finite strings such that the equivalence class of a finite history γ_n is determined by the state of E after having read γ_n).

Definition 3.2 Let N be a nondeterministic automaton on AxK-trees. Let R be a regular equivalence relation on the set H^* of finite histories. A run ρ of N on an AxK-tree t is (K,R)-uniform if, at every pair of vertices u and v of t which have the same K-projection and are such that the histories of the copies of N at u and v are R-equivalent ; N makes the same transition.

Theorem II (Uniformization) Given a nondeterministic automaton N on AxK-trees, one can effectively construct a regular equivalence relation R such that if N has an accepting run on an AxK-tree t then N has a (K,R)-uniform accepting run on t.

This theorem is weaker then the theorem of Gurevich and Harrington [2] in that the choice made at a vertex may depend on the complete history of the automaton present at the vertex, but seems to us to be simpler exactly for this reason. Our proof is by an induction on the number of states of N and uses Muchnik's ingeneous idea of decomposing the behaviour of N into a composite of the behaviours of the machines N_i obtained respectively by converting the state q_{i+1} into a dead-end marker. As a consequence we have

Theorem III Given an alternating automaton M on K-trees on can effectively construct a nondeterministic automaton P on K-trees which is equivalent to M.

Corollary 1 The class of languages accepted by alternating automata is closed under projection.

We now have a simple algorithm for the emptiness problem as follows. Given an alternating automaton M on K-trees we use Theorem III to construct a nondeterministic automaton P on K-trees which is equivalent to M. As usual, we may projetct to the case of an alphabet with one letter. Now the computation trees are the same for an alternating automaton on K-trees labelled by a single letter a and the automaton considered as an alternating automaton on the line with state set LxQ. Again applying Theorem III, we find a nondeterministic automaton J on the natural numbers such that the language $L(J)$ is empty if and only if $L(M)$ is empty. This gives a "collapsing" of the tree onto the line.

Corollary 2 There is an effective construction which, when given a sentence ϕ of the monadic theory of the k-ary tree, associates a sentence ϕ' of the monadic theory of the natural numbers such that ϕ' is true if and only if ϕ is true.

BIBLIOGRAPHY

1. A.K. Chandra D. Kozen and L. Stockmeyer, Alternation, J.A.C.M. 28 (1981), 114-133.

2. Y. Gurevich and L. Harrington, Trees, Automata and Games, Proc. ACM STOC May 1982, 60-65.

3. Y.N. Moschovakis, Descriptive Set Theory, North Holland, Amsterdam, 1980.

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