INFORMATION AND CONTROL **13**, 484–491 (1968)

The Minimalization of Tree Automata

WALTER S. BRAINERD

Department of Mathematics, U. S. Naval Postgraduate School, Monterey, California 93940

A tree automaton is a system $\langle Q, f_1, \cdots, f_k, F\rangle$ where Q is a set of states, $f_1, \cdots, f_k$ are operations on Q of arbitrary finite index, and $F \subseteq Q$ is a set of final states. The input to a tree automaton is a tree structure and thus the behavior of a tree automaton is a set of trees. These automata are generalizations of ordinary automata, in which all f's are unary. An algorithm for constructing a minimal tree automaton is given.

1. INTRODUCTION

Büchi (1966) has observed that finite automata are algebras in which the set of states is the carrier and all operators are unary except the initial state, which is 0-ary. The advantage of this representation is that many algebraic results may be immediately applied to automata theory. This approach has allowed the recent generalizations made by Doner (1967) and Thatcher and Wright (1966), who have investigated automata as algebras in which the operators may be of arbitrary finite index. This is an interesting generalization because these automata may be interpreted as machines which accept tree structures as input.

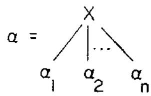
In this paper we consider the minimalization problem for tree automata. These results were a part of the author's doctoral thesis, Brainerd (1967). Arbib and Give'on (1968) independently investigated the minimalization problem using a somewhat different approach.

2. TREES

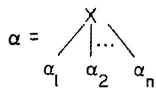
The trees considered in this paper should more properly be called labelled ordered trees. The labels will be members of some alphabet A, but only trees of a certain type will be considered; therefore, we first make the following definition:

DEFINITION 2.1. A *stratified alphabet*, Gorn (1966), is a pair $\langle A, \sigma\rangle$, where A is a finite set of symbols and $\sigma: A \rightarrow N = \{0, 1, 2, \cdots\}$. Let $A_n = \sigma^{-1}(n)$.

DEFINITION 2.2. α is a *tree* over A iff 1) $\alpha = x \in A_0$ or 2)



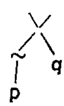
where $x \in A_n$ and $\alpha_1, \alpha_2, \cdots, \alpha_n$ are trees over A. Note that the stratification of a label at each node must equal the number of branches going down from the node. For typographical reasons, the tree



will always be represented by the string $\alpha_1\alpha_2 \cdots \alpha_n x$, the Polish postfix notation for the tree. Let A^T be the set of trees over A. Note that $A^T = \varnothing$ if $A_0 = \varnothing$. (Thatcher and Wright (1966) use prefix notation and call A^T the set of terms over A. The use of postfix notation allows a tree automaton to read a tree from left to right.)

DEFINITION 2.3. The *depth* of a tree is defined as follows:
1) $d(x) = 0, x \in A_0$
2) $d(\alpha_1 \cdots \alpha_n x) = 1 + \max\{d(\alpha_i) \mid 1 \leqq i \leqq n\}$.

Example 2.4. Let $A = \{\vee, \sim, p, q\}$, $\sigma(\vee) = 2$, $\sigma(\sim) = 1$, $\sigma(p) = \sigma(q) = 0$. In this case A^T is the set of formulas of propositional calculus which contain at most two statement letters. The tree



which in this paper will be written $p \sim q\vee$ would usually be written $\sim p \vee q$ using infix notation.

It should be noted that trees are generalizations of strings. Let $A = \{\#, x_1, \cdots, x_k\}$ where $\sigma(\#) = 0$ and $\sigma(x_i) = 1$, $1 \leqq i \leqq k$. Then any tree $\#\alpha \in A^T$ may be identified with the string $\alpha \in (A - \{\#\})^*$.

3. TREE AUTOMATA

DEFINITION 3.1. Let $\langle A, \sigma\rangle$ be a stratified alphabet, where $A = \{x_1, x_2, \cdots, x_k\}$. A *tree automaton* over A is a system $M = \langle Q, f_1, f_2, \cdots, f_k, F\rangle$ where
1) Q is a set of *states*;
2) $f_i : Q^{\sigma(x_i)} \rightarrow Q$, $1 \leqq i \leqq k$;

and

3) $F \subseteq Q$ is a set of *final* states.

M will be called a *finite* automaton if Q is finite.

Notation. If $x = x_i$, then f_x means f_i. The functions $f_1, f_2, \cdots, f_k$ are called *transition functions.*

We now indicate how each tree automaton accepts or rejects a tree in A^T. The functions f_i induce a response function ρ: $A^T \to Q$, which, with F, defines a subset of A^T.

DEFINITION 3.2. The *response function* ρ of a tree automaton M is defined as follows:

1) if $x \in A_0$, $\rho(x) = f_x$

2) if $x \in A_n$, $n > 0$, $\rho(\alpha_1\alpha_2 \cdots \alpha_n x) = f_x(\rho(\alpha_1), \rho(\alpha_2), \cdots, \rho(\alpha_n))$.

DEFINITION 3.3. $T(M) = \{\alpha \in A^T \mid \rho(\alpha) \in F\}$. $T(M)$ is the *behavior* of M and we say that M *accepts* $T(M)$. M_1 and M_2 are *equivalent* iff $T(M_1) = T(M_2)$.

Example 3.4. Let $A = \{\vee, \sim, p, q\}$ as in example 2.4. Let $M = \langle Q, f_\vee, f_\sim, f_p, f_q, F\rangle$, where $Q = \{X, Y, Z\}$, $f_p = X$, $f_\vee(X, X) = f_\vee(X, Y) = Y$, $f_\vee = Z$ in all other cases, $f_q = Z$, $f_\sim = Z$ in all cases, and $F = \{X, Y\}$. It may be verified that $T(M) = \{p^{n+1} \vee^n \mid n \geqq 0\}$.

Note that if $A = \{\#, x_1, \cdots, x_k\}$, where $\sigma(\#) = 0$ and $\sigma(x_i) = 1$ $1 \leqq i \leqq k$, then $\langle Q, f_\# = q_0, f_1, \cdots, f_k, F\rangle$ is an ordinary automaton, accepting strings $\#\alpha$, $\alpha \in (A - \{\#\})^*$. In this sense q_0 should be thought of as the response to the first symbol $\#$, rather than a start state.

It is felt that something is lost if trees are viewed simply as strings when written in postfix or any other linear form. For example, if the trees accepted by the automaton in example 3.4 are written in prefix form, the automaton accepts the set $\{(p\vee)^n p \mid n \geqq 0\}$, which is accepted by an ordinary finite automaton, whereas the set $\{p^{n+1} \vee^n \mid n \geqq 0\}$ is not.

The purpose of this paper is to give an algorithm for constructing a minimal automaton. We are usually interested in finite automata; however, free automata, together with the concepts of homomorphism and congruence provide the key to the minimalization problem.

DEFINITION 3.5. Let $\Sigma \subseteq A^T$. $M^*(\Sigma) = \langle A^T, f_1^*, \cdots, f_k^*, \Sigma\rangle$ is the *free automaton* with respect to Σ, where $f_x^*(\alpha_1, \cdots, \alpha_n) = \alpha_1 \cdots \alpha_n x$ if $\sigma(x) = n$.

The identity function is the response of a free automaton and $T(M^*(\Sigma)) = \Sigma$. Thus, every subset of A^T is accepted by some automaton.

4. HOMOMORPHISMS AND CONGRUENCES

DEFINITION 4.1. Let $M_i = \langle Q_i, f_1^i, \cdots, f_k^i, F_i \rangle$, $i = 1, 2$, be tree automata over A. $h: Q_1 \to Q_2$ is a *homomorphism* of M_1 onto M_2 and we write $M_1 \Rightarrow M_2$ if and only if 1) $h(Q_1) = Q_2$, 2) $\sigma(x) = n$ implies $h(f_x^1(X_1, \cdots, X_n)) = f_x^2(hX_1, \cdots, hX_n)$, for all $X_1, \cdots, X_n \in Q_1$ and all $x \in A_n$, and 3) $X \in F_1$ iff $hX \in F_2$. A 1–1 homomorphism of M_1 onto M_2 is an *isomorphism*.

LEMMA 4.2. *Suppose $M_1 \Rightarrow M_2$. Let ρ_1, ρ_2 be the response functions of M_1, M_2. Then $\forall \alpha \in A^T$, $h(\rho_1(\alpha)) = \rho_2(\alpha)$.*

Proof. The proof is by induction on the depth of α.

1) $d(\alpha) = 0$ implies $\alpha = x \in A_0$ implies $h(\rho_1(\alpha)) = h(f_x^1) = f_x^2 = \rho_2(\alpha)$.
2) Let $\alpha = \alpha_1 \cdots \alpha_n x$, where $\alpha(x) = n > 0$ and assume that for each α' such that $d(\alpha') < d(\alpha)$, $h(\rho_1(\alpha')) = \rho_2(\alpha')$. Since $d(\alpha_i) < d(\alpha)$, $i = 1, 2, \cdots, n$, $h(\rho_1(\alpha)) = h[f_x^1(\rho_1(\alpha_1), \cdots, \rho_1(\alpha_n))] = f_x^2[h(\rho_1(\alpha_1)), \cdots, h(\rho_1(\alpha_n))] = f_x^2[\rho_2(\alpha_1), \cdots, \rho_2(\alpha_n)] = \rho_2(\alpha_1 \cdots \alpha_n x) = \rho_2(\alpha)$.

LEMMA 4.3. *If $M_1 \Rightarrow M_2$, then $T(M_1) = T(M_2)$.*

Proof. $\alpha \in T(M_1)$ iff $\rho_1(\alpha) \in F_1$ iff $h(\rho_1(\alpha)) \in F_2$ iff $\rho_2(\alpha) \in F_2$ iff $\alpha \in T(M_2)$.

DEFINITION 4.4. $M = \langle Q, f_1, \cdots, f_k, F \rangle$ is *reduced* iff $\forall X \in Q$, $\exists \alpha \in A^T \ni \rho(\alpha) = X$.

LEMMA 4.5. *If $M = \langle Q, f_1, \cdots, f_k, F \rangle$ is reduced and $T(M) = \Sigma$, then $M^*(\Sigma) \Rightarrow M$ and the response ρ of M is the homomorphism. That is, all reduced automata accepting Σ are homomorphic images of the free automaton with respect to Σ.*

Proof. 1) $\rho(A^T) = Q$ since M is reduced. 2) Let $\sigma(x) = n$, then $\rho[f_x^*(\alpha_1, \cdots, \alpha_n)] = \rho(\alpha_1 \cdots \alpha_n x) = f_x[\rho(\alpha_1), \cdots, \rho(\alpha_n)]$. 3) $\alpha \in \Sigma$ iff $\alpha \in T(M)$ iff $\rho(\alpha) \in F$.

DEFINITION 4.6. Let $M = \langle Q, f_1, \cdots, f_k, F \rangle$ be a reduced tree automaton. $\sim$ is a *congruence* on M iff 1) $\sim$ is an equivalence relation on Q, 2) if $\sigma(x) = n$ and $X_i \sim Y_i$, $i = 1, \cdots, n$, then $f_x(X_1, \cdots, X_n) \sim$

$f_x(Y_1, \cdots, Y_n)$, i.e., $\sim$ is compatible with each transition function f_x, and 3) if $X \sim Y$ then $X \in F \equiv Y \in F$.

DEFINITION 4.7. Let $\sim$ be a congruence on $M = \langle Q, f_1, \cdots, f_k, F\rangle$. Let $X/\sim = \{Y \in Q \mid Y \sim X\}$, $Q/\sim = \{X/\sim \mid X \in Q\}$, $f_x/\sim(X_1/\sim, \cdots, X_n/\sim) = f_x(X_1, \cdots, X_n)/\sim$ and $F/\sim = \{X/\sim \mid X \in F\}$. Note that $f_x/\sim$ and $F/\sim$ are well-defined since $\sim$ is a congruence. $M/\sim = \langle Q/\sim, f_1/\sim, \cdots, f_k/\sim, F/\sim\rangle$ is the *quotient* of M modulo $\sim$.

LEMMA 4.8. *If $\sim$ is a congruence on M, then $M \Rightarrow M/\sim$.*

Proof. Let $h(X) = X/\sim$.

LEMMA 4.9. *If h is a homomorphism of M_1 onto M_2 (both reduced), then $X \sim Y$* iff *$hX = hY$ is a congruence on M_1 and $M_1/\sim \Leftrightarrow M_2$.*

The proof is immediate from the definitions of homomorphism, congruence and quotient.

These results may be summarized as follows:

THEOREM 4.10. $\{M \mid T(M) = \Sigma, M \text{ reduced}\} = \{M \mid M^*(\Sigma) \Rightarrow M\} = \{M^*(\Sigma)/\sim \mid \sim \text{ a congruence on } M^*(\Sigma)\}$.

COROLLARY 4.11. *$\Sigma \subseteq A^T$ is accepted by a finite automaton* iff *Σ is the union of some of the equivalence classes of an equivalence relation on A^T which has finite index and which is compatible with the functions f_x^*.*

5. THE MINIMALIZATION ALGORITHM

Given a finite tree automaton, we wish to construct an automaton with the same behavior having the fewest states. We believe that if the material in this paper is specialized to the case of unary transition functions, the result will be a nice presentation of the minimalization problem for ordinary automata. The results and proofs will appear somewhat less complicated in the special case.

We begin by eliminating all states which are the response of no tree. This can be done effectively because of the following lemma. It is a generalization of a result for ordinary automata, Büchi (1966) and Rabin and Scott (1959), and is proved in Doner (1967) and in Thatcher and Wright (1966).

LEMMA 5.1. *Let M be a finite tree automaton with q states. For each state $X \in Q$, if there is a tree α such that $\rho(\alpha) = X$, then there is a tree β such that $d(\beta) < q$ and $\rho(\beta) = X$.*

LEMMA 5.2. *Given a finite tree automaton $M = \langle Q, f_1, \cdots, f_k, F\rangle$ over A, one can effectively construct an equivalent reduced automaton.*

Proof. Let $Q' = \{X \in Q \mid \exists \alpha \in A^T \ni \rho(\alpha) = X\}$. Q' may be found effectively by Lemma 5.1. Let $M' = \langle Q', f_1 \mid (Q')^{\sigma(x_1)}, \cdots, f_k \mid (Q')^{\sigma(x_k)}, F \cap Q'\rangle$. M' is a tree automaton whose response is identical to the response of M. $\alpha \in T(M)$ iff $\rho(\alpha) \in F$ iff $\rho(\alpha) \in F \cap Q'$ iff $\alpha \in T(M')$, hence M and M' are equivalent.

In the remainder of this paper, all automata are assumed to be reduced. Note that a free automaton is reduced.

We now define an easily computed congreunce $\simeq$, such that if M is finite, $M/\simeq$ will be minimal.

DEFINITION 5.3. Let $M = \langle Q, f_1, \cdots, f_k, F\rangle$ be a tree automaton. Let

1) $X \sim_0 Y$ iff $X \in F \equiv Y \in F$
2) $X \sim_{m+1} Y$ iff $X \sim_m Y$ and $\forall x \in A \ni \sigma(x) = n$, $\forall X_1, \cdots, X_n \in Q$, $[f_x(X, X_2, \cdots, X_n) \sim_m f_x(Y, X_2, \cdots, X_n) \wedge f_x(X_1, X, \cdots, X_n) \sim_m f_x(X_1, Y, \cdots, X_n) \wedge \cdots \wedge f_x(X_1, X_2, \cdots, X) \sim_m f_x(X_1, X_2, \cdots Y)]$.

Define $X \simeq Y$ iff $\forall m$, $X \sim_m Y$.

Note that $X \sim_{m+1} Y$ implies $X \sim_m Y$ and $\sim_m = \sim_{m+1}$ implies $\sim_{m+1} = \sim_{m+2}$. Thus, if Q is finite, $\exists m_0 \ni \sim_{m_0} = \sim_{m_0+j}$, $j = 1, 2, \cdots$. In this case $\simeq = \sim_{m_0}$.

LEMMA 5.4. *$\simeq$ is a congruence on any automaton M.*

Proof. To show that $\simeq$ is an equivalence relation is a straightforward but somewhat tedious exercise left to the reader. To show that $\simeq$ is compatible with the transition functions, suppose $X_i \simeq Y_i$, $1 \leqq i \leqq n$. Let m be arbitrary; then $X_i \sim_{m+1} Y_i$, $1 \leqq i \leqq n$, hence $f_x(X_1, X_2, \cdots, X_n) \sim_m f_x(Y_1, X_2, \cdots, X_n) \sim_m f_x(Y_1, Y_2, \cdots, X_n) \sim_m \cdots \sim_m f_x(Y_1, Y_2, \cdots, Y_n)$, by the definition of $\sim_{m+1}$. Thus $\forall m$, $f_x(X_1, \cdots, X_n) \sim_m f_x(Y_1, \cdots, Y_n)$, since $\sim_m$ is transitive. Finally, $X \simeq Y$ implies $X \sim_0 Y$ implies $X \in F \equiv Y \in F$.

LEMMA 5.5. *If $\sim$ is a congruence on $M = \langle Q, f_1, \cdots, f_k, F\rangle$, then $X \sim Y$ implies $X \simeq Y$,* i.e., *any congruence on M is a refinement of $\simeq$. Also, $M/\sim \Rightarrow M/\simeq$.*

Proof. We show $X \sim Y$ implies $X \sim_m Y$ for all m, by induction on m.

1) If $X \sim Y$, then $X \in F \equiv Y \in F$, since $\sim$ is a congruence. Hence $X \sim_0 Y$.

2) Suppose $X \sim Y$ implies $X \sim_m Y$. Then $\forall x \in A_n$, $X_1, \cdots, X_n \in Q$, $[f_x(X, X_2, \cdots, X_n) \sim f_x(Y, X_2, \cdots, X_n) \wedge \cdots \wedge f_x(X_1, X_2, \cdots, X) \sim f_x(X_1, X_2, \cdots, Y)]$, since $\sim$ is a congruence. By the induction hypothesis, $\forall x \in A_n$, $\forall X_1 \cdots X_n \in Q$, $[f_x(X, X_2, \cdots, X_n) \sim_m f_x(Y, X_2, \cdots, X_n) \wedge \cdots \wedge f_x(X_1, X_2, \cdots, X) \sim_m f_x(X_1, X_2, \cdots, Y)$. Hence $X \sim_{m+1} Y$ by definition of $\sim_{m+1}$.

$h(X/\sim) = X/\simeq$ is the homomorphism of $M/\sim$ onto $M/\simeq$.

LEMMA 5.6. *If $T(M) = \Sigma$, then $M^*(\Sigma)/\simeq \Leftrightarrow M/\simeq$, where $\simeq$ is the congruence of Definition*5.3.

Proof. $M^*(\Sigma) \Rightarrow M$ and there is a congreunce $\sim$ such that $M = M^*(\Sigma)/\sim$ by Theorem 4.10. By Lemma 5.4, $M = M^*(\Sigma)/\sim \Rightarrow M^*(\Sigma)/\simeq$. Using Lemma 5.4 again, $M^*(\Sigma)/\simeq \Rightarrow M/\simeq$, since $M^*(\Sigma)/\simeq$ is a quotient of M. On the other hand $M^*(\Sigma) \Rightarrow M \Rightarrow M/\simeq$. Hence, using Lemma 5.4 again, $M/\simeq \Rightarrow M^*(\Sigma)/\simeq$, since $M/\simeq$ is a quotient of $M^*(\Sigma)$.

THEOREM 5.7. *If M is a finite reduced tree automaton and $T(M) = \Sigma$, then $M/\simeq$ is the unique (up to isomorphism) automaton which has the smallest number of states and accepts Σ.*

Proof. Let M' be any reduced automaton such that $T(M') = \Sigma$. By Lemma 5.5, $M' \Rightarrow M'/\simeq \Leftrightarrow M^*(\Sigma)/\simeq \Leftrightarrow M/\simeq$. Thus $M' \Rightarrow M/\simeq$, which means that either M' is isomorphic to $M/\simeq$ or M' has more states than $M/\simeq$.

ALGORITHM 5.8. Let M be a finite tree automaton. To construct the minimal automaton equivalent to M:

1) Find the reduced automaton M' equivalent to M using Lemma 5.2.
2) Construct congruences $\sim_0, \sim_1, \cdots$ on M' in accordance with Definition 5.3 until $\sim_m = \sim_{m+1}$.
3) Construct $M'/\sim_m$, which is the desired minimal automaton.

RECEIVED: July 22, 1968

REFERENCES

ARBIB, M. A., AND Y. GIVE'ON (1968), Algebraic automata I: parallel programming as a prolegomenon to the categorical approach. *Inform. Control* **12**, 331–345.

BRAINERD, W. S. (1967), Tree generating systems and tree automata. Doctoral thesis, Purdue University.

BÜCHI, J. R. (1966), Algebraic theory of feedback in discrete systems, part I. *In* "Automata Theory," (E. R. Caianiello, ed.). Academic Press, New York.

DONER, J. E. (1967), Tree acceptors and some of their applications. System Development Corporation, Scientific Report No. 8.

GORN, S. (1966), Explicit definitions and linguistic dominoes. Presented at the Systems and Computer Science Conference, University of Western Ontario, London, Ontario.

RABIN, M. O., AND D. SCOTT (1959), Finite automata and their decision problems. *IBM J. Research and Development.* **3**, 114–125.

THATCHER, J. W., AND J. B. WRIGHT (1966), Generalized finite automata theory with an application to a decision problem of second-order logic. IBM Research Paper RC-1713.