

**Three Families of Maximally
Nondeterministic Automata**

by

Tom Verhoeff

87/10

June, 1987

COMPUTING SCIENCE NOTES

This is a series of notes of the Computing Science Section of the Department of Mathematics and Computing Science Eindhoven University of Technology. Since many of these notes are preliminary versions or may be published elsewhere, they have a limited distribution only and are not for review. Copies of these notes are available from the author or the editor.

**Eindhoven University of Technology
Department of Mathematics and Computing Science
P.O. Box 513
5600 MB EINDHOVEN
The Netherlands
All rights reserved
editors: prof.dr.M.Rem
 prof.dr.K.M.van Hee.**

Three Families of Maximally Nondeterministic Automata

TOM VERHOEFF

Department of Mathematics and Computing Science
Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
mcvax.UUCP!eutrc3!wstomv

April 1987

ABSTRACT

For every nondeterministic finite state machine there exists an equivalent deterministic automaton. Such an equivalent automaton can be obtained by the subset construction. An automaton may be called *maximally nondeterministic* when the subset construction yields a minimal automaton. This note presents three nice families of maximally nondeterministic automata together with some variations. The languages concerned are all prefix-closed.

CONTENTS

- 0. Introduction
 - 1. Deterministic and maximally nondeterministic automata
 - 2. A simple family
 - 3. A family based on the doubling construction
 - 4. A family with only two symbols
 - 5. Concluding remarks
- References

0. INTRODUCTION

Some circuit synthesis techniques based on Trace Theory (cf. [0], [1]) require the conversion of a nondeterministic automaton into an equivalent deterministic one. This conversion may cause an increase in the size of the automaton. It is interesting to know how bad the increase can be. This note presents a couple of worst case examples.

There are many ways to define automata (cf. [2], [3], [4]). In this note an *automaton* is a quadruple $\langle Q, A, \delta, q_0 \rangle$, where

Q is a finite set of *states*,

A is a finite set of *symbols*,

δ is a mapping of $Q \times A$ to the power set of Q , and

q_0 is in Q .

The set A is called the *alphabet* of the automaton, δ its *transition function*, and q_0 its *initial state*.

Let N be an automaton, $N = \langle Q, A, \delta, q_0 \rangle$. We also write qN for Q , aN for A , δ_N for δ , and iN for q_0 .

The *state transition graph* of N , or state graph for short, is the following rooted labeled directed graph G . G has vertex set Q . There is an arrow labeled b , $b \in A$, from vertex q to vertex r if and only if $r \in \delta(q, b)$. The root of G is q_0 . A labeled arrow is also called a *transition* of the automaton. The set of transitions is denoted by xN ; it can be defined as a subset of $Q \times A \times Q$. We are only interested in its size as a measure of the automaton's complexity. We have

$$\#xN = (\text{SUM } q, b : q \in Q \wedge b \in A : \#\delta(q, b)). \quad (0.0)$$

Notice that the alphabet of an automaton may contain symbols that do not occur as labels in its state graph. An automaton is, therefore, not uniquely determined by its state graph. For the automata that we shall encounter each symbol of the alphabet occurs in at least one transition.

The *trace* of a path in the state graph is the sequence of symbols that occur as labels on its constituent arrows. The *trace set* of a state is the set of all traces of paths that begin in that state. The trace set of a state is thus a subset of $(aN)^*$. The trace set of an automaton is the trace set of its initial state. Two automata are called *equivalent* if and only if their trace sets are equal.

Although the above definitions in terms of the state graph are straightforward, it is also convenient to express them more directly. Therefor the transition function δ is extended to sets of states and traces of symbols as follows. For X a subset of Q and t a trace in A^* we define $\hat{\delta}(X, t)$ by

$$\hat{\delta}(X, \epsilon) = X,$$

$$\hat{\delta}(X, tb) = (\cup q : q \in \hat{\delta}(X, t) : \delta(q, b)) \quad \text{for } b \in A,$$

where ϵ denotes the empty trace. The extended transition function $\hat{\delta}$ has some important properties. Let X and Y be subsets of Q , q a state in Q , t and u traces over A , and b a symbol in A . Then we have

$$\hat{\delta}(\emptyset, t) = \emptyset, \tag{0.1}$$

$$\hat{\delta}(\{q\}, b) = \delta(q, b), \tag{0.2}$$

$$\hat{\delta}(X, tu) = \hat{\delta}(\hat{\delta}(X, t), u), \text{ and} \tag{0.3}$$

$$\hat{\delta}(X \cup Y, t) = \hat{\delta}(X, t) \cup \hat{\delta}(Y, t). \tag{0.4}$$

In state graph terminology, $\hat{\delta}(\{q\}, t)$ is the set of states that can be reached from state q via a path with trace t . Hence, there exists a path in the state graph with trace t starting in state q if and only if $\hat{\delta}(\{q\}, t) \neq \emptyset$. Thus, the trace set of state q is the set

$$\{t \in (aN)^* \mid \hat{\delta}(\{q\}, t) \neq \emptyset\}.$$

Technical Remark. The kind of automata that we have defined are also known as

- 0) nondeterministic finite state machines,
- 1) with exactly one initial state,
- 2) without epsilon transitions,
- 3) that are not necessarily completely specified (since δ may give \emptyset as image), and
- 4) of which all states are final states.

A consequence of points 1) and 4) is that the trace sets of our automata are nonempty and prefix-closed. The latter means that for each trace in the trace set all its prefixes are also in the trace set.

(End of Technical Remark)

1. DETERMINISTIC AND MAXIMALLY NONDETERMINISTIC AUTOMATA

Automaton N is *deterministic* if and only if

$$(\forall q, b : q \in Q \wedge b \in A : \#\delta(q, b) \leq 1).$$

In terms of the state graph this means that all outgoing arrows of a vertex have distinct labels. Hence, for each vertex there is at most one path with a given trace that starts in this vertex. The well-known *subset construction*, when applied to N , yields a deterministic automaton D that is equivalent to N . We recall that

$$qD = \{X \mid \emptyset \neq X \subseteq qN\},$$

$$aD = aN,$$

$$\delta_D(X, b) = \begin{cases} \{\hat{\delta}_N(X, b)\} & \text{if } \hat{\delta}_N(X, b) \neq \emptyset \\ \emptyset & \text{otherwise} \end{cases} \quad \text{for } X \in \mathbf{q}D \text{ and } b \in \mathbf{a}D,$$

$$\mathbf{i}D = \{\mathbf{i}N\},$$

and therefore $\#\mathbf{q}D = 2^{\#\mathbf{q}N} - 1$. D is also called the *subset automaton* of N .

Technical Remark. Notice that we have excluded the empty set from $\mathbf{q}D$. It could have been included, but then it should be a nonfinal state—otherwise D would not be equivalent to N . Since no path starting in it would lead to a final state, it can be omitted as far as the trace set is concerned. That is what we have done.

(End of Technical Remark)

The following relation between the extended transition functions of N and D holds.

$$\hat{\delta}_D(\{X\}, t) = \begin{cases} \{\hat{\delta}_N(X, t)\} & \text{if } \hat{\delta}_N(X, t) \neq \emptyset \\ \emptyset & \text{otherwise} \end{cases} \quad \text{for } X \in \mathbf{q}D \text{ and } t \in (\mathbf{a}N)^*. \quad (1.0)$$

Automaton M is *minimal* if and only if M is deterministic and $\#\mathbf{q}M \leq \#\mathbf{q}D$ for all deterministic automata D equivalent to M . All minimal automata that are equivalent to automaton N trivially have the same number of states (by the way, they are even isomorphic). That number is denoted by $\mathbf{m}N$. If N is deterministic then obviously $\mathbf{m}N \leq \#\mathbf{q}N$. From the subset construction we, therefore, know that in general

$$\mathbf{m}N \leq 2^{\#\mathbf{q}N} - 1. \quad (1.1)$$

Automaton N is called *maximally nondeterministic* if and only if $\mathbf{m}N = 2^{\#\mathbf{q}N} - 1$. Hence, N is maximally nondeterministic if and only if its subset automaton is minimal.

In the following sections we present three families of maximally nondeterministic automata. In each family there is for every integer n , $n \geq 1$, one maximally nondeterministic automaton with n states.

The first family is interesting because its definition and correctness proof are very simple. But it uses unnecessarily large alphabets and transition sets (exponential in the number of states). The second family is of interest because it is based on a general "doubling" construction. The sizes of its alphabets and transition sets are linear and quadratic in the number of states respectively. The third family is the compulsive optimizer's delight since it is very economic: all its automata use two symbols and the number of transitions is twice the number of states.

Before embarking on the definitions of these families we characterize minimal automata and maximally nondeterministic automata. Well-known (cf. Corollary 5.3 in [3]) is this

Theorem. Deterministic automaton D is minimal if and only if

- 0) all states of D are *accessible*, i.e., in the state graph of D there is a path from the root to every state,

and

- 1) all states of D are *distinguishable*, i.e., every two distinct states have different trace sets.

(End of Theorem)

The accessibility condition for automaton D can also be expressed as

$$(\mathbf{A} \ q : q \in \mathbf{q}D : (\mathbf{E} \ t : t \in (\mathbf{a}D)^* : q \in \hat{\delta}_D(\{\mathbf{i}D\}, t))).$$

Application of relation (1.0) yields the following

Corollary. Automaton N is maximally nondeterministic if and only if

$$(\mathbf{A} \ X : \emptyset \neq X \subseteq \mathbf{q}N : (\mathbf{E} \ t : t \in (\mathbf{a}N)^* : X = \hat{\delta}_N(\{\mathbf{i}N\}, t))) \quad (1.2)$$

and

$$(\mathbf{A} \ X, Y : \emptyset \neq X \subseteq \mathbf{q}N \wedge \emptyset \neq Y \subseteq \mathbf{q}N \wedge X \neq Y \\ : \{t \in (\mathbf{a}N)^* \mid \hat{\delta}_N(X, t) \neq \emptyset\} \neq \{t \in (\mathbf{a}N)^* \mid \hat{\delta}_N(Y, t) \neq \emptyset\}). \quad (1.3)$$

(End of Corollary)

When proving that automaton N is maximally nondeterministic we shall speak of the accessibility and distinguishability of nonempty subsets of $\mathbf{q}N$ when dealing with (1.2) and (1.3) respectively. Actually these notions refer to the subset automaton of N .

2. A SIMPLE FAMILY

This section presents our first family of maximally nondeterministic automata. It is a simple family in the sense that the proofs are straightforward.

For integer $n, n \geq 1$, we define automaton N by

$$\mathbf{q}N = \{k \mid 0 \leq k < n\},$$

$$\mathbf{a}N = \{b_X \mid \emptyset \neq X \subseteq \mathbf{q}N\} \cup \{c_k \mid k \in \mathbf{q}N\},$$

$$\delta_N(k, b_X) = \begin{cases} X & \text{if } k = 0 \\ \emptyset & \text{otherwise} \end{cases} \quad \text{for } b_X \in \mathbf{a}N.$$

$$\delta_N(k, c_j) = \begin{cases} \{k\} & \text{if } k = j \\ \emptyset & \text{otherwise} \end{cases} \quad \text{for } c_j \in \mathbf{a}N, \text{ and}$$

$$\mathbf{i}D = 0.$$

Hence, we have

$$\# \mathbf{q}N = n ,$$

$$\# \mathbf{a}N = 2^n + n - 1, \text{ and}$$

$$\# \mathbf{x}N = n 2^{n-1} + n.$$

The automata in this family are maximally nondeterministic.

Proof. We show that the accessibility and distinguishability conditions are satisfied.

First we deal with accessibility. Let X be a nonempty subset of $\mathbf{q}N$ (a state of the subset automaton). Hence, b_X is a symbol of N . We show that X is accessible (in the subset automaton) via the trace consisting of the single symbol b_X :

$$\begin{aligned} & \hat{\delta}_N(\{0\}, b_X) \\ = & \quad \{ \text{property (0.2) of } \hat{\delta} \} \\ & \delta_N(0, b_X) \\ = & \quad \{ \text{definition of } \delta_N \} \\ & X \end{aligned}$$

Next comes distinguishability. Let X and Y be distinct nonempty subsets of $\mathbf{q}N$. Without loss of generality assume $j \in X \setminus Y$. We show that the trace consisting of the single symbol c_j is in the trace set of X but not in that of Y (w.r.t. the subset automaton). Hence, X and Y are distinguishable states (in the subset automaton). We derive

$$\begin{aligned} & \hat{\delta}_N(X, c_j) \\ = & \quad \{ \text{definition of } \hat{\delta}, \text{ or properties (0.2) and (0.4) } \\ & \quad (\cup k : k \in X : \delta_N(k, c_j)) \} \\ = & \quad \{ \text{definition of } \delta_N, \text{ using } j \in X \} \\ & \{ j \} \\ \neq & \quad \{ \text{set calculus } \} \\ & \emptyset \end{aligned}$$

and similarly

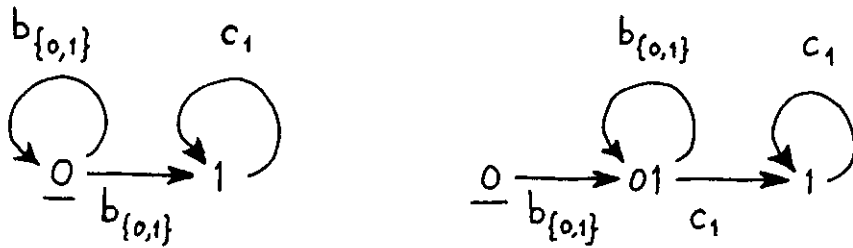
$$\begin{aligned} & \hat{\delta}_N(Y, c_j) \\ = & \quad \{ \text{as above, using } j \notin Y \} \\ & \emptyset \end{aligned}$$

(End of Proof)

Since these automata have so many symbols and transitions, we do not give examples of their state graphs. One variation that slightly reduces the number of symbols and transitions, is obtained by omitting symbols $b_{\{k\}}$ for $0 \leq k < n$ and c_0 , and all transitions

involving these symbols. The set $\{k\}$, $0 < k < n$, is now accessible via the trace $b_{qN}c_k$. For the purpose of distinguishability, b_{qN} can take over the role of c_0 .

For $n = 2$ the state graph of this variation is given below, together with the state graph of its subset automaton. The initial state is underlined. In the state graph of the subset automaton the states are denoted by the concatenation of their elements.



3. A FAMILY BASED ON THE DOUBLING CONSTRUCTION

In this section we present a construction that can be applied to any automaton N to yield an automaton N' with

$$\#qN' = \#qN + 1, \tag{3.0}$$

$$\#aN' = \#aN + 2, \tag{3.1}$$

$$\#xN' = \#xN + \#qN + \#aN + 2, \text{ and} \tag{3.2}$$

$$mN' = 2 \cdot mN + 1. \tag{3.3}$$

For an obvious reason we call it the *doubling construction*. From (3.3) and

$$2(2^n - 1) + 1 = 2^{n+1} - 1 \tag{3.4}$$

it follows that N' is maximally nondeterministic if and only if N is maximally nondeterministic.

Using the doubling construction we can obtain a family of maximally nondeterministic automata in the following way. Define N_1 as $\langle \{0\}, \emptyset, \emptyset, 0 \rangle$ and take for N_{n+1} , with $n \geq 1$, the result of the doubling construction applied to N_n . N_1 is deterministic and minimal, since it has only one state. Furthermore, because $2^1 - 1 = 1$, automaton N_1 is maximally nondeterministic. Hence, by mathematical induction on n all automata N_n , for $n \geq 1$, are maximally nondeterministic. Their parameters are

$$\#qN_n = n,$$

$$\#aN_n = 2n - 2, \text{ and}$$

$$\#xN_n = 3n(n - 1)/2.$$

Doubling Construction. Let N be an automaton. Then N' is defined by

$$\mathbf{q}N' = \mathbf{q}N \cup \{q_0\} \quad \text{where } q_0 \notin \mathbf{q}N \text{ is a fresh state,}$$

$$\mathbf{a}N' = \mathbf{a}N \cup \{x, y\} \quad \text{where } x \notin \mathbf{a}N \text{ and } y \notin \mathbf{a}N \text{ are fresh symbols,}$$

$$\delta_{N'}(p, b) = \delta_N(p, b) \quad \text{for } p \in \mathbf{q}N \text{ and } b \in \mathbf{a}N,$$

$$\delta_{N'}(p, x) = \emptyset \quad \text{for } p \in \mathbf{q}N,$$

$$\delta_{N'}(p, y) = \{p\} \quad \text{for } p \in \mathbf{q}N,$$

$$\delta_{N'}(q_0, b) = \{q_0\} \quad \text{for } b \in \mathbf{a}N,$$

$$\delta_{N'}(q_0, x) = \{\mathbf{i}N, q_0\},$$

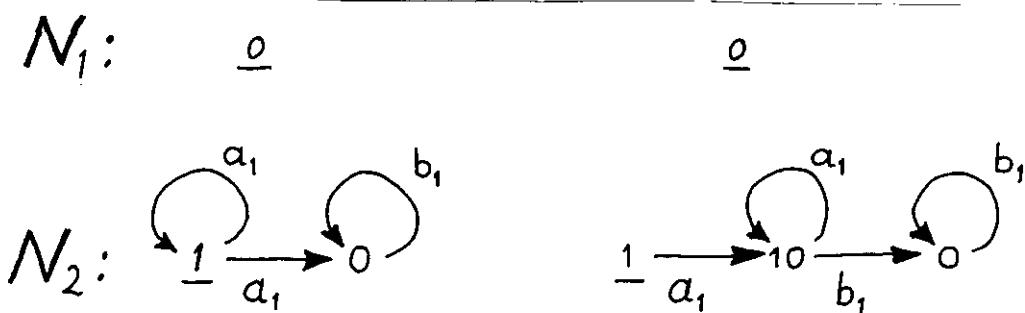
$$\delta_{N'}(q_0, y) = \emptyset, \text{ and}$$

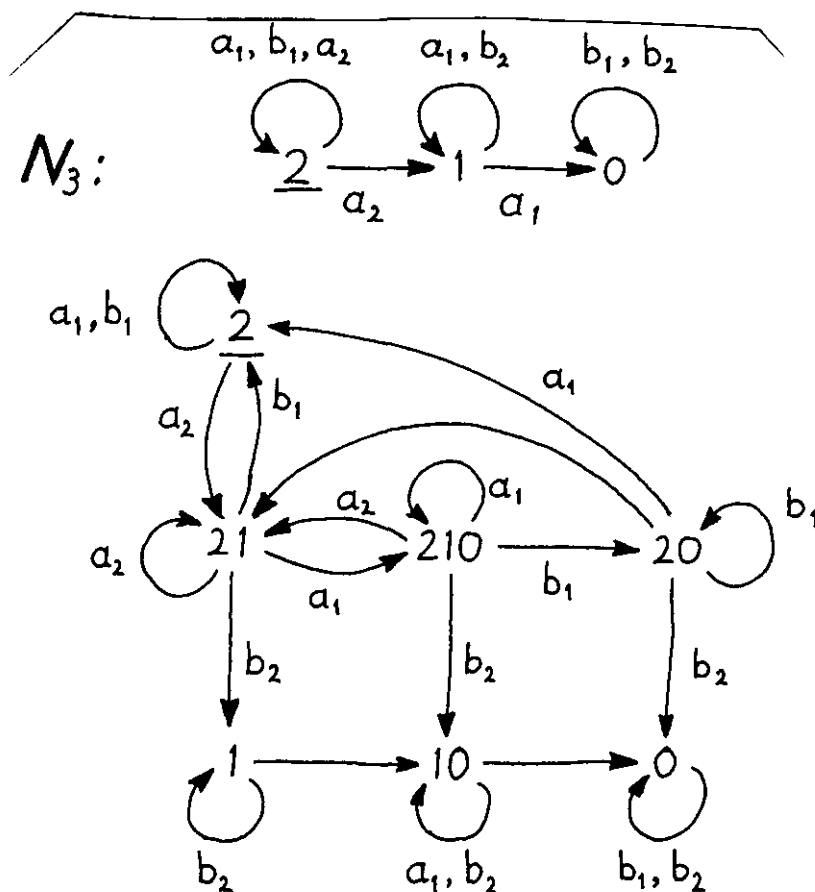
$$\mathbf{i}N' = q_0.$$

The state graph of N' can be obtained from that of N by adding (i) a transition labeled y from each state to itself, (ii) the new initial state q_0 , (iii) a transition labeled x from q_0 to itself and to $\mathbf{i}N$, and (iv) for each symbol of $\mathbf{a}N$ a transition labeled with that symbol from q_0 to itself.

(End of Doubling Construction)

Examples. We give the state graphs of the automata N_n as discussed above, for $1 \leq n \leq 3$, and also of their subset automata. For the new initial state of N_{n+1} we have chosen n ; the fresh symbols of N_{n+1} are a_n and b_n , playing the roles of x and y respectively. An arrow labeled with a list of symbols abbreviates multiple "parallel" arrows, each labeled with a symbol from that list. Notice that an isomorphic version of N_2 appeared in the preceding section as a variation on the simple family.





(End of Examples)

That the doubling construction fulfills claims (3.0) through (3.2) is obvious from its definition. We shall now prove that it also satisfies (3.3) in the case that N is maximally nondeterministic. The general case is left as an exercise to the reader.

Proof. Let N be a maximally nondeterministic automaton and let N' be the automaton obtained by applying the doubling construction to N . We use q_0 , x , and y to denote the corresponding objects called for in the doubling construction.

On account of (3.4) it suffices to prove that N' is maximally nondeterministic. We intend to do this by showing the accessibility and distinguishability conditions for the subset automaton of N' . But first we notice the following properties of the extended transition function of N' . They derive from the properties of $\hat{\delta}$ mentioned in Section 0 and the definition of $\hat{\delta}_{N'}$. For X a (possibly empty) subset of qN and t a trace over aN we have

$$\hat{\delta}_{N'}(X, t) = \hat{\delta}_N(X, t), \tag{3.5}$$

$$\hat{\delta}_{N'}(X, x) = \emptyset, \tag{3.6}$$

$$\hat{\delta}_{N'}(X, y) = X. \tag{3.7}$$

$$\hat{\delta}_{N'}(\{q_0\}, t) = \{q_0\}, \quad (3.8)$$

$$\hat{\delta}_{N'}(\{q_0\}, x) = \{iN, q_0\}, \text{ and} \quad (3.9)$$

$$\hat{\delta}_{N'}(\{q_0\}, y) = \emptyset. \quad (3.10)$$

Now we prove that the accessibility condition is satisfied (cf. (1.2)). Let X be a nonempty subset of $\mathbf{q}N'$. We distinguish two cases: $q_0 \in X$ and $q_0 \notin X$.

Case $q_0 \in X$. If $X = \{q_0\}$ then the empty trace trivially suffices. Suppose $X \setminus \{q_0\}$ is nonempty, and therefore a nonempty subset of $\mathbf{q}N$. On account of (1.2) applied to N , which is assumed maximally nondeterministic, let t be a trace over $\mathbf{a}N$ such that $\hat{\delta}_N(\{iN\}, t) = X \setminus \{q_0\}$. We show that the trace xt accesses X :

$$\begin{aligned} & \hat{\delta}_{N'}(\{q_0\}, xt) \\ = & \quad \{ \text{property (0.3) of } \hat{\delta} \} \\ & \hat{\delta}_{N'}(\hat{\delta}_{N'}(\{q_0\}, x), t) \\ = & \quad \{ (3.9) \} \\ & \hat{\delta}_{N'}(\{iN, q_0\}, t) \\ = & \quad \{ \text{set calculus and property (0.4) of } \hat{\delta} \} \\ & \hat{\delta}_{N'}(\{iN\}, t) \cup \hat{\delta}_{N'}(\{q_0\}, t) \\ = & \quad \{ (3.5) \text{ and } (3.8) \text{ using that } t \in (\mathbf{a}N)^* \} \\ & \hat{\delta}_N(\{iN\}, t) \cup \{q_0\} \\ = & \quad \{ \text{definition of } t \} \\ & X \setminus \{q_0\} \cup \{q_0\} \\ = & \quad \{ \text{set calculus using that } q_0 \in X \} \\ & X \end{aligned}$$

Case $q_0 \notin X$. On account of the previous case $X \cup \{q_0\}$ is accessible. Let t be a trace over $\mathbf{a}N'$ such that $\hat{\delta}_{N'}(\{q_0\}, t) = X \cup \{q_0\}$. We show that X is accessed via the trace ty :

$$\begin{aligned} & \hat{\delta}_{N'}(\{q_0\}, ty) \\ = & \quad \{ \text{property (0.3) of } \hat{\delta} \} \\ & \hat{\delta}_{N'}(\hat{\delta}_{N'}(\{q_0\}, t), y) \\ = & \quad \{ \text{definition of } t \} \\ & \hat{\delta}_{N'}(X \cup \{q_0\}, y) \\ = & \quad \{ \text{property (0.4) of } \hat{\delta} \} \\ & \hat{\delta}_{N'}(X, y) \cup \hat{\delta}_{N'}(\{q_0\}, y) \\ = & \quad \{ (3.7), (3.10), \text{ and set calculus} \} \\ & X \end{aligned}$$

And now we prove the distinguishability condition (cf. (1.3)). Let X and Y be distinct nonempty subsets of $\mathfrak{q}N'$. We consider three cases: $q_0 \notin X \cup Y$, $q_0 \in X \div Y$, and $q_0 \in X \cap Y$.

Case $q_0 \notin X \cup Y$. Then X and Y are distinct nonempty subsets of $\mathfrak{q}N$. On account of (1.3) and the assumption that N is maximally nondeterministic there is a trace over $\mathfrak{a}N$ that distinguishes (the trace sets of) X and Y in (the subset automaton of) N . From (3.5) we see that the same trace reveals the distinguishability of X and Y in N' .

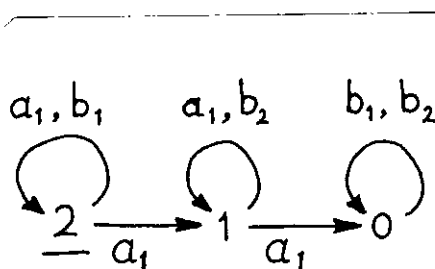
Case $q_0 \in X \div Y$. Without loss of generality assume $q_0 \in Y$, and thus $X \subseteq \mathfrak{q}N$. Using property (0.4) of $\hat{\delta}$, the definition of $\hat{\delta}_{N'}$, and (3.6) for $Y \setminus \{q_0\}$, one obtains $\hat{\delta}_{N'}(Y, y) = \{iN, q_0\}$. Together with (3.6) for X , this shows that the trace consisting of the single symbol x distinguishes X and Y .

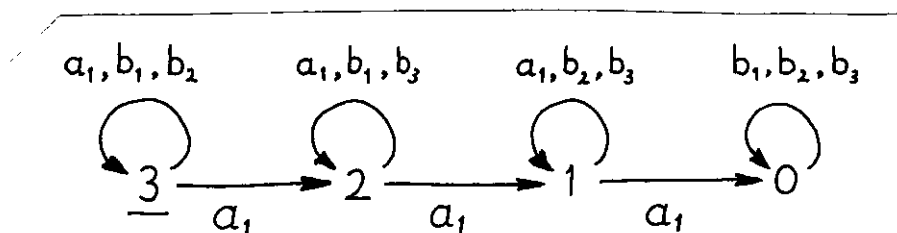
Case $q_0 \in X \cap Y$. If either X or Y equals $\{q_0\}$, then the trace y distinguishes X and Y on account of (3.7) and (3.10). Otherwise, $X \setminus \{q_0\}$ and $Y \setminus \{q_0\}$ are nonempty and distinct. Hence, the first case applies. Say trace t , $t \in (\mathfrak{a}N)^*$, distinguishes them. Then trace yt distinguishes X and Y , since $\hat{\delta}_{N'}(X, y) = X \setminus \{q_0\}$ and similarly for y .

(End of Proof)

In general the doubling construction requires two new symbols for each additional state. If, however, it is repeatedly applied to N_1 —as was done above—then one new symbol suffices. In this special case one can take the "new" x equal to the analogous symbol of the previous automaton. In that way one obtains a family of maximally nondeterministic automata for which the number of symbols equals the number of states (except for the single-state automaton, which has an empty alphabet). Below we give a condensed proof of this assertion.

Examples. We give the state graphs with 3 and 4 states obtained by carrying out the above variation.





(End of Examples)

The alternative family of automata can also be described nonrecursively. For that description it is convenient to rename the symbol a_1 as b_0 . The transition function δ of the n -state automaton is then given by

$$\delta(k, b_j) = \begin{cases} \{k, k-1\} & \text{if } k \neq j = 0 \\ \{k\} & \text{if } k \neq j > 0 \\ \emptyset & \text{if } k = j \end{cases}$$

where $0 \leq k < n$ and $0 \leq j < n$. This allows us to deduce the following properties of the extended transition function. For integer m , $0 \leq m < n$, we have

$$\hat{\delta}(\{n-1\}, (b_0)^m) = \{k \mid n-m-1 \leq k < n\}, \tag{3.11}$$

and for X a subset of states and j a state, $j > 0$,

$$\hat{\delta}(X, b_j) = X \setminus \{b_j\}. \tag{3.12}$$

Since $n-1$ is the initial state of the n -state automaton, the accessibility condition (1.2) is now easily verified. Let X be a nonempty set of states. Apply (3.11) with $m = n-2$ if $0 \notin X$ and with $m = n-1$ if $0 \in X$. Follow this by a suitable sequence of applications of (3.12) to remove unwanted states and thereby obtain X .

To show that the distinguishability condition (1.3) holds, consider two distinct nonempty sets of states X and Y . Without loss of generality assume $X \setminus Y$ is nonempty. Define trace t as the sequence of symbols b_j where the indexes form a permutation of Y , such that b_0 is at the end if $0 \in Y$. Trace t distinguishes X and Y on account of (3.12) and $\delta(\{0\}, b_0) = \emptyset$.

The automata of the alternative family can be reduced further by omitting those transitions labeled with symbol b_0 ($= a_1$) that go from state k to itself, where k is not the initial state. This reduction does not affect properties (3.11) and (3.12) and the fact that $\delta(\{0\}, b_0) = \emptyset$. The resulting automata have only one state that violates the condition for being deterministic, viz. the initial state.

4. A FAMILY WITH ONLY TWO SYMBOLS

A moment's thought will convince the reader that a maximally nondeterministic automaton with more than one state requires an alphabet with at least two symbols. The family presented in this section shows that two symbols suffice for any number of states.

For integer $n, n \geq 2$, automaton N is defined by

$$qN = \{k \mid 0 \leq k < n\},$$

$$aN = \{a, b\},$$

$$\delta_N(0, a) = \{0, 1\},$$

$$\delta_N(0, b) = \emptyset,$$

$$\delta_N(k, a) = \{(k+1) \bmod n\} \quad \text{for } 0 < k < n,$$

$$\delta_N(k, b) = \{k\} \quad \text{for } 0 < k < n, \text{ and}$$

$$iN = 0.$$

We do not bother to make an explicit choice for a single-state automaton of this family. It shall not concern us here. The automata are not indexed with n , since n will be fixed throughout this section. We shall also omit the subscript N on δ_N and $\hat{\delta}_N$.

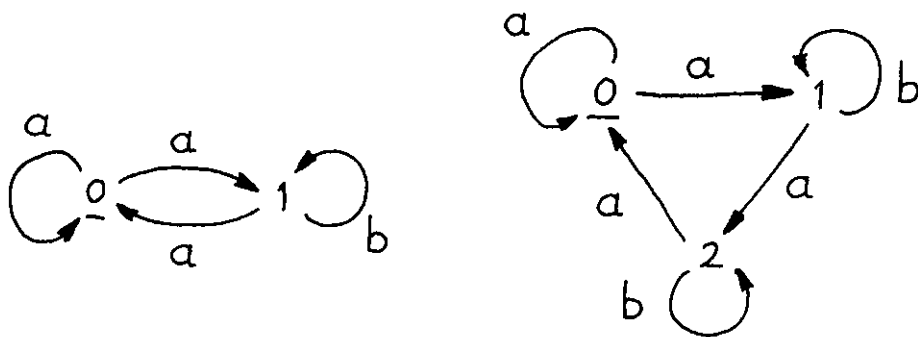
The family has the following parameters.

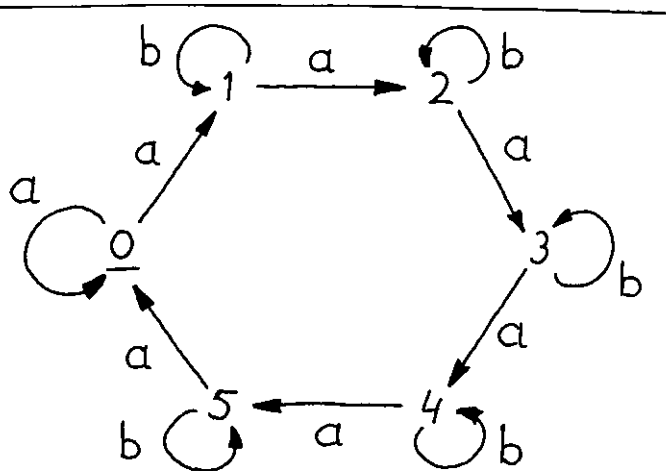
$$\#qN = n,$$

$$\#aN = 2, \text{ and}$$

$$\#xN = 2n.$$

Examples. We give the state graphs for the automata with 2, 3, and 6 states.





(End of Examples)

Notice that the initial state is the only state that violates the condition for being deterministic: it has two outgoing arrows with label a . Also notice that the initial state is the only state that does not have outgoing arrows with both labels a and b .

We use a special notation for sets of states—viz. *bit vectors*—to prove that these automata are maximally nondeterministic.

The characteristic function over Q of the set X , $X \subseteq Q$, is the mapping f from Q into $\{0, 1\}$ defined by

$$f(q) = \begin{cases} 0 & \text{if } q \notin X \\ 1 & \text{if } q \in X \end{cases} \quad \text{for } q \in Q.$$

A bit vector is just a way of writing the characteristic function of a set. We are interested in subsets of qN . The bit vector of X , $X \subseteq qN$, is the string obtained by concatenating the images of X 's characteristic function over qN , while traversing the arguments (elements of qN) in increasing order. These bit vectors all have length n . We shall apply standard abbreviations regarding strings to bit vectors as well. For example, the empty set has bit vector 0^n and the initial state of N 's subset automaton, i.e. the set $\{0\}$, has bit vector 10^{n-1} .

In what follows t denotes a trace over aN , u , v , and w denote bit vectors, and x , y , and z denote bits (bit vectors of length 1). The lengths of u , v , and w are implicitly assumed to be such that the bit vector in which they appear has the proper length.

The automata of the two-symbol family are maximally nondeterministic.

Proof. Let n be an integer, $n \geq 2$. We consider the automaton N with n states of this family. From its definition and the properties of $\hat{\delta}$ as mentioned in Section 0, one can derive the following properties of N 's extended transition function.

$$\hat{\delta}(1uy, a) = 11u, \quad (4.0)$$

$$\hat{\delta}(0uy, a) = y0u, \text{ and} \quad (4.1)$$

$$\hat{\delta}(xv, b) = 0v. \quad (4.2)$$

When taking $y = 1$, properties (4.0) and (4.1) can be combined into

$$\hat{\delta}(v1, a) = 1v. \quad (4.3)$$

From properties (4.0), (4.1), and (4.2) also follow

$$\hat{\delta}(vy, ab) = 0v, \text{ and} \quad (4.4)$$

$$\hat{\delta}(xuy, ba) = y0u. \quad (4.5)$$

First, we prove by mathematical induction on the length of u (denoted by k) that the nonempty set of states with bit vector $u1^{n-k}$ is accessible. Taking $k = n$ then shows that the accessibility condition (1.2) is satisfied.

Base $u = \epsilon$. Repeated application of (4.0) yields

$$\hat{\delta}(10^{n-1}, a^{n-1}) = 1^n,$$

which shows that 1^n is accessible.

Step $u = xv$. On account of the induction hypothesis let t be a trace over aN with

$$\hat{\delta}(01^{n-1}, t) = v1^{n-k+1}$$

Since $k \leq n$ we can apply (4.3) and (4.4) to $v1^{n-k+1}$. Together with (0.3) and the definition of t this gives

$$\hat{\delta}(10^{n-1}, ta) = 1v1^{n-k}, \text{ and}$$

$$\hat{\delta}(10^{n-1}, tab) = 0v1^{n-k}.$$

Distinguishing the cases $x = 1$ and $x = 0$, this shows that $xu1^{n-k}$ is accessed via traces ta and tab respectively.

Finally, we prove that the distinguishability condition (1.3) holds. Let u and v be the bit vectors of two distinct nonempty sets of states. Because the sets are nonempty, u can be written as $0^{n-1-k}1u_0$, and v as $0^{n-1-l}1v_0$, where k and l are the lengths of u_0 and v_0 respectively. We distinguish two cases: $k \neq l$ and $k = l$.

Case $k \neq l$. Without loss of generality assume $k > l$. We claim that the trace $(ab)^{l+1}$ belongs to the trace set of u but not to that of v . Write u_0 as u_1u_2 , where the length of u_2 is $l+1$. We compute

$$\begin{aligned} & \hat{\delta}(u, (ab)^{l+1}) \\ = & \quad \{ \text{rewriting } u \} \end{aligned}$$

$$\begin{aligned}
 & \hat{\delta}(0^{n-1-k} 1u_1u_2, (ab)^{l+1}) \\
 = & \quad \{ \text{repeated application of string calculus, (0.3), and (4.4), using } k > l \} \\
 & 0^{n-k+l} 1u_1 \\
 \neq & \quad \{ \text{string calculus} \} \\
 & 0^n
 \end{aligned}$$

and

$$\begin{aligned}
 & \hat{\delta}(v, (ab)^{l+1}) \\
 = & \quad \{ \text{rewriting } v \} \\
 & \hat{\delta}(0^{n-1-l} 1v_0, (ab)^{l+1}) \\
 = & \quad \{ \text{repeated application of string calculus, (0.3), and (4.4)} \} \\
 & 0^n
 \end{aligned}$$

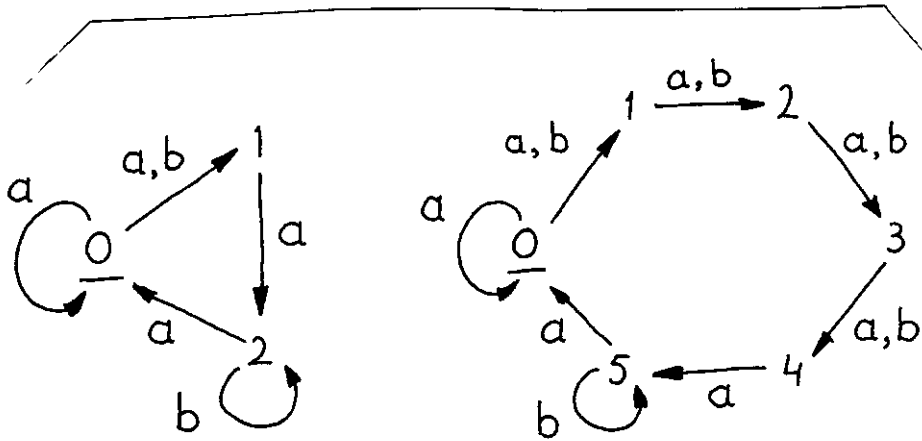
Case $k = l$. Since u and v are distinct we can rewrite u_0 as u_1yu_2 and v_0 as v_1zv_2 , such that $y \neq z$ and u_2 has the same length as v_2 . Let m be the length of u_2 . Consider the trace $(ab)^m ba$. Writing xw for $0^{n-1-k+m} 1$, we compute

$$\begin{aligned}
 & \hat{\delta}(u, (ab)^m ba) \\
 = & \quad \{ \text{rewriting } u, \text{ string calculus, and (0.3)} \} \\
 & \hat{\delta}(\hat{\delta}(0^{n-1-k} 1u_1yu_2, (ab)^m), ba) \\
 = & \quad \{ m \text{ applications of (4.4), and definition of } xw \} \\
 & \hat{\delta}(xwu_1y, ba) \\
 = & \quad \{ (4.5) \} \\
 & y0wu_1
 \end{aligned}$$

Similarly we obtain $\hat{\delta}(v, (ab)^m ba) = z0wv_1$. Because $y \neq z$ the previous case applies to $y0wu_1$ and $z0wv_1$ (they indeed represent distinct nonempty sets). If these states can be distinguished by trace t , then it follows that trace $(ab)^m bat$ distinguishes u and v .

(End of Proof)

The two-symbol automata discussed above are a variation on a family suggested by Berry Schoenmakers. For completeness' sake the state graphs of his automata with 3 and 6 states are depicted below. The proofs for these automata can be given along the same lines, but they are somewhat more complicated (especially accessibility).



5. CONCLUDING REMARKS

Jo Ebergen brought the question whether upper bound (1.1) can be attained under my attention. Everyone that I asked about it told me that maximally nondeterministic automata—attaining this upper bound—were indeed known to exist. They could not, however, give me any references. Exercise 2.2.27 of [2], for example, poses the problem, but no solution is provided. Finally, I set out to construct them myself. This resulted in the first two families. Berry Schoenmakers put me on the scent of the third family. Only afterwards did I hit upon Theorem 1.10 of [5], which exhibits a family of maximally nondeterministic automata with three symbols. References [3] and [4] are also good introductions to automata theory, but they don't mention the problem. Nevertheless both have somehow influenced my presentation.

The class of nondeterministic automata considered in this note is restricted in the sense that the corresponding trace sets are nonempty and prefix-closed. This is the class that is most prominent in the circuit synthesis methods. It was imaginable that the upper bound (1.1) could be attained by more general nondeterministic automata but not within the restricted class. It turns out, however, that the upper bound is attained even in the restricted class.

Another restricted class that is of interest consists of the automata obtained from deterministic ones by concealment of certain symbols. Concealed symbols can be modeled by transitions without symbols (also called *epsilon transitions*; cf. [4], [0]). Upper bound (1.1) cannot be attained within this class, as is shown in [6]. The tightest upper bound for this class is

$$mN \leq 3 \cdot 2^{*qN-2} - 1$$

for automata with at least two states. The upperbound is attained by the second variation discussed in Section 3 by concealing the symbol on the transition from the initial state to the "next" state.

I have given the definitions and correctness proofs in such detail because it is very easy to come up with sloppy suggestions that afterwards turn out to be ill-defined or plainly wrong. Thanks are due to a number of colleagues for pointing out shortcomings of a—much shorter—draft version.

REFERENCES

- [0] J.L.A. van de Snepscheut, *Trace Theory and VLSI Design*, Lecture Notes in Computer Science, Vol. 200, Springer, 1985.
- [1] J.C. Ebergen, *Translating Programs into Delay-Insensitive Circuits*, Dissertation, Eindhoven University of Technology, Dept. of Math. and Comp. Sci., The Netherlands, 1987 (in preparation).
- [2] A.V. Aho and J.D. Ullman, *The Theory of Parsing and Compiling*, Vol. 1, Prentice-Hall, 1972.
- [3] S. Eilenberg, *Automata, Languages, and Machines*, Vol. A, Academic Press, 1974.
- [4] J.E. Hopcroft and J.D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, 1979.
- [5] B.A. Trakhtenbrot and Ya.M. Barzdin', *Finite Automata: Behavior and Synthesis*, North-Holland, 1973.
- [6] H.M.J.L. Schols, *The Maximum Number of States after Projection*, Computing Science Notes, No. 87/06, Eindhoven University of Technology, Dept. of Math. and Comp. Sci., The Netherlands, April 1987.

COMPUTING SCIENCE NOTES

In this series appeared :

No.	Author(s)	Title
85/01	R.H. Mak	The formal specification and derivation of CMOS-circuits
85/02	W.M.C.J. van Overveld	On arithmetic operations with M-out-of-N-codes
85/03	W.J.M. Lemmens	Use of a computer for evaluation of flow films
85/04	T. Verhoeff H.M.J.L. Schols	Delay insensitive directed trace structures satisfy the foam rubber wrapper postulate
86/01	R. Koymans	Specifying message passing and real-time systems
86/02	G.A. Bussing K.M. van Hee M. Voorhoeve	ELISA, A language for formal specifications of information systems
86/03	Rob Hoogerwoord	Some reflections on the implementation of trace structures
86/04	G.J. Houben J. Paredaens K.M. van Hee	The partition of an information system in several parallel systems
86/05	Jan L.G. Dietz Kees M. van Hee	A framework for the conceptual modeling of discrete dynamic systems
86/06	Tom Verhoeff	Nondeterminism and divergence created by concealment in CSP
86/07	R. Gerth L. Shira	On proving communication closedness of distributed layers

86/08	R. Koymans R.K. Shyamasundar W.P. de Roever R. Gerth S. Arun Kumar	Compositional semantics for real-time distributed computing (Inf.&Control 1987)
86/09	C. Huizing R. Gerth W.P. de Roever	Full abstraction of a real-time denotational semantics for an OCCAM-like language
86/10	J. Hooman	A compositional proof theory for real-time distributed message passing
86/11	W.P. de Roever	Questions to Robin Milner - A responder's commentary (IFIP86)
86/12	A. Boucher R. Gerth	A timed failures model for extended communicating processes
86/13	R. Gerth W.P. de Roever	Proving monitors revisited: a first step towards verifying object oriented systems (Fund. Informatica IX-4)
86/14	R. Koymans	Specifying passing systems requires extending temporal logic
87/01	R. Gerth	On the existence of sound and complete axiomatizations of the monitor concept
87/02	Simon J. Klaver Chris F.M. Verberne	Federatieve Databases
87/03	G.J. Houben J.Paredaens	A formal approach to distri- buted information systems
87/04	T.Verhoeff	Delay-insensitive codes - An overview

- 87/05 R.Kuiper Enforcing non-determinism via linear time temporal logic specification.
- 87/06 R.Koymans Temporele logica specificatie van message passing en real-time systemen (in Dutch).
- 87/07 R.Koymans Specifying message passing and real-time systems with real-time temporal logic.
- 87/08 H.M.J.L. Schols The maximum number of states after projection.
- 87/09 J. Kalisvaart
L.R.A. Kessener
W.J.M. Lemmens
M.L.P. van Lierop
F.J. Peters
H.M.M. van de Wetering Language extensions to study structures for raster graphics.
- 87/10 T.Verhoeff Three families of maximally nondeterministic automata.
- 87/11 P.Lemmens Eldorado ins and outs.
Specifications of a data base management toolkit according to the functional model.