Theoretical Computer Science 38 (1985) 133-136
North-Holland

NOTE

SUCCINCT REPRESENTATION OF REGULAR LANGUAGES BY BOOLEAN AUTOMATA II

Ernst LEISS

Department of Computer Science, University of Houston, Houston, TX 77004, U.S.A.

Communicated by M. Nivat
Received October 1984

Abstract. Boolean automata are a generalization of finite automata in the sense that the next state (the result of the transition function, given a state and a letter) is not just a single state (deterministic automata) or a set of states (nondeterministic automata) but a boolean function of the states. Boolean automata accept precisely the regular languages; also, they correspond in a natural way to certain language equation involving complementation as well as to sequential networks. In a previous note we showed that for every $n \geq 1$, there exists a boolean automaton $\boldsymbol{B}_n$ with n states such that the smallest deterministic automaton for the same language has 2^{2^n} states. In the present note we will show a precisely attainable lower bound on the succinctness of representing regular languages by boolean automata; namely, we will show that, for every $n \geq 1$, there exists a reduced automaton $\boldsymbol{D}_n$ with n states such that the smallest boolean automaton accepting the same language has also n states.

Notation

A boolean automaton (for more details and proofs, see [2, 3, 5]) is a quintuple $\boldsymbol{B} = (A, Q, \tau, f^0, F)$ where A is the input alphabet, Q is the finite nonempty set of states, $\tau: Q \times A \to B_Q$ is the transition function, B_Q denoting the free boolean algebra generated by Q, $f^0 \in B_Q$ is the initial function and $F \subseteq Q$ is the set of final states. The transition function is extended to $B_Q \times A^*$ in the usual way. We define a relation $=_F$ as follows: Let $\delta_i = 1$ if $q_i \in F$, otherwise $\delta_i = 0$. For any $f \in B_Q$, $f =_F \alpha$ iff $f(\delta_1, \ldots, \delta_n) = \alpha$ where $Q = \{q_1, \ldots, q_n\}$ and $\alpha \in \{0, 1\}$. A word $w \in A^*$ is accepted by $\boldsymbol{B}$ if and only if $\tau(f^0, w) =_F 1$; the set of all such words is denoted by $L(\boldsymbol{B})$.

If $f^0 \in Q$ and $\tau(q_i, a_j) \in Q$ for all i, j, then $\boldsymbol{B}$ is a deterministic (finite) automaton; if f^0 is a union of states and all the $\tau(q_i, a_j)$ are unions of states, $\boldsymbol{B}$ is a nondeterministic automaton. We will always assume that our boolean automata are connected, i.e., for no $P \subsetneq Q$, $\{\tau(f^0, w) \mid w \in A^*\}$ is a subset of $B_P \subsetneq B_Q$. It is known [2, 3] that $L(\boldsymbol{B})$ is always a regular language. Furthermore, the derived deterministic automaton $\boldsymbol{A_B}$ accepts precisely $L(\boldsymbol{B})$; $\boldsymbol{A_B}$ is defined as follows: $\boldsymbol{A_B} = (A, P, \mu, f^0, G)$ where $P = \{\tau(f^0, w) \mid w \in A^*\} \subseteq B_Q$, $G = \{f \in P \mid f =_F 1\}$, and $\mu(\tau(f^0, w), a) = \tau(f^0, wa)$ for all $w \in A^*$, $a \in A$. Clearly, if $\boldsymbol{B}$ has n states, $\boldsymbol{A_B}$ can have no more than 2^{2^n} states.

The reverse N^ρ of a (connected) deterministic finite automaton $N=(A, Q, \tau, q_0, F)$ is defined as follows. For any $w \in A^*$, let $Q_w = \{q \in Q \mid \tau(q, w) \in F\}$. Then $N^\rho = (A, Q, \mu, p_0, G)$ where $P = \{p \mid p = Q_w$ for some $w \in A^*\}$, $p_0 = F$, $G = \{p \in P \mid q_0 \in p\}$ and $\mu(p, a) = \{q \in Q \mid \tau(q, a) \in p\}$ for $p \in P$, $a \in A$. N^ρ is always reduced, has at most 2^n states if N has n states, and the language accepted by N^ρ is precisely the reverse of the language accepted by N, $(L(N))^\rho = L(N^\rho)$ (see [1]).

1. Introduction

It is well known that for every $n \geq 1$ there exists a *nondeterministic* finite automaton N_n with n states such that the smallest *deterministic* finite automaton for $L(N_n)$ has 2^n states. In other words, there exist regular languages of nondeterministic complexity n whose deterministic complexity is 2^n. In [6] we posed the same problem for *boolean* automata. More specifically, we showed that for every $n \geq 1$ there is a boolean automaton B_n with n states such that the reduced automaton for $L(B_n)$ has exactly the attainable maximum, namely 2^{2^n} states. Thus there are languages of boolean complexity n whose deterministic complexity is 2^{2^n}. Furthermore, an alphabet of two letters suffices; for three-letter alphabets the result had been proven earlier by Kozen [3]. Thus, boolean automata provide a very succinct representation of regular languages, although not as succinct as extended regular expressions [7].

In this note we are interested in the 'worst case' of this representation, namely we ask what is the largest number of states of boolean automata which is required for representing regular languages accepted by reduced automata with n states, as a function of n. This question has of course an analogue for nondeterministic finite automata. There it is known that for every $n \geq 1$ there exist languages of deterministic complexity n which are also of nondeterministic complexity n. For example, $a^{n-1}a^*$ over the alphabet $\{a\}$ is such a language. It is clear that it is of deterministic complexity n. To see that no nondeterministic automaton with fewer than n states can accept this language, consider any shortest path from an initial state to a final state in such an automaton. This immediately leads to a contradiction. That the situation for boolean automata is not so easy, directly follows from [6, Theorem 1] which states that the reverse of every language of deterministic complexity n is of boolean complexity $\lceil \log_2 m \rceil$. Consequently, languages over a single-letter alphabet or single words will suffer a logarithmic reduction in complexity when going from deterministic to boolean.

In the following we will prove nevertheless that there are languages whose deterministic complexity equals their boolean complexity. Again an alphabet of two letters suffices.

2. The Theorem

We prove the following theorem.

Theorem. *For every $n \geq 1$ there exists a reduced automaton $\boldsymbol{D}_n$ with n states such that any boolean automaton $\boldsymbol{B}$ which accepts precisely $L(\boldsymbol{D}_n)$ has at least n states.*

Proof. For $n=1$ the result holds trivially. Thus assume $n \geq 2$. Consider the deterministic finite automaton $\boldsymbol{D}_n = (\{0, 1\}, \{0, 1, \ldots, n-1\}, \tau_n, 1, F_n)$ with $F_n = \{i \mid 0 \leq i \leq n-1 \text{ and } i \text{ even}\}$ and the transition function given by

$$\tau_n(i, 0) = (i+1) \bmod n, \qquad \tau_n(i, 1) = \begin{cases} i, & 0 \leq i \leq n-1, \\ n-1, & i = n-2, n-1. \end{cases}$$

From [6] we will use that this automaton has the property that $(\boldsymbol{D}_n)^\rho$ has exactly 2^n states. It is known that the reverse of any deterministic automaton is reduced provided the given automaton is connected (see [1]). From this it follows first of all that $\boldsymbol{D}_n$ is reduced; for assume there were another deterministic automaton $\boldsymbol{D}'$ with fewer than n, say m, states which accepts $L(\boldsymbol{D}_n)$. Then the reverse $(\boldsymbol{D}')^\rho$ would have at most 2^m states which is strictly less than 2^n. This is a contradiction to the fact that $(\boldsymbol{D}_n)^\rho$ is reduced; $\boldsymbol{D}_n$ is reduced for all $n \geq 2$.

The next fact we need is the result due to Kozen ([4]; see also [6]) that the reverse of any language accepted by an n-state boolean automaton can be accepted by a deterministic automaton with no more than 2^n states. Now assume that there exists a boolean automaton $\boldsymbol{B}$ with fewer than n states, say s, which accepts $L(\boldsymbol{D}_n)$. By Kozen's theorem it follows that the reverse of $L(\boldsymbol{D}_n)$ can be accepted by a deterministic automaton with at most 2^s states. This is in contradiction to the fact that $(\boldsymbol{D}_n)^\rho$ has precisely 2^n ($>2^s$) states and is reduced. Thus the theorem follows. □

From the proof of this result we can derive two corollaries.

Corollary 1. *Every regular language of deterministic complexity n whose reverse language has deterministic complexity 2^n is also of boolean complexity n.*

Corollary 2. *Every regular language of deterministic complexity n whose boolean complexity is n is also of nondeterministic complexity n.*

This last observation immediately follows from the fact that every nondeterministic automaton can be considered a boolean automaton.

References

[1] J.A. Brzozowski, Canonical regular expressions and minimal state graph for definite events, in: *Mathematical Theory of Automata*, Symposia Series **12** (Polytechnic Institute of Booklyn, Brooklyn, 1963) 529–561.

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

[3] D. Kozen, On parallelism in Turing machines, *Proc. 17th Ann. Symp. on Foundations of Computer Science* (1976) 89–97.

[4] D. Kozen, Private communication, April 1980.
[5] E. Leiss, On generalized language equations, *Theoret. Comput. Sci.* **14** (1981) 63-77.
[6] E. Leiss, Succinct representation of regular languages by boolean automata, *Theoret. Comput. Sci.* **13** (1981) 323-330.
[7] A.R. Meyer and L. Stockmeyer, Nonelementary word problems in automata and logic, *Proc. AMS Symp. on Complexity of Computation*, 1973.