TWO-WAY DETERMINISTIC FINITE AUTOMATA ARE EXPONENTIALLY MORE SUCCINCT THAN SWEEPING AUTOMATA *

Silvio MICALI
University of California, Berkeley, CA 94720, U.S.A.

Received 23 February 1980; revised version received 2 October 1980

Complexity, size of description, nondeterminism, finite automata, two-way finite automata, sweeping automata

1. Problem set up

It is well known that one-way deterministic finite automata (1dfa), two-way deterministic finite automata (2dfa), one-way nondeterministic finite automata (1nfa) and two-way nondeterministic finite automata (2nfa) all recognize the same set of languages: the regular languages. However, how do they compare with each other with respect to the size of their description?

We will adopt as a measure for the size of the description of a finite automaton the number of its states. The main problem, still open, has ben posed in [1] by Sakoda and Sipser:

For every 2nfa M, is there an equivalent 2dfa with only polynomially more states than M?

A partial answer to the above question has been given in [2] by Sipser. He introduced the notion of a sweeping automaton as follows:

Definition. A sweeping automaton (sa) is a 2dfa which can halt or change the direction of its head motion only at the end of the input tape.

He proved that:

For all n there is a language B_n which is accepted by an n-state 1nfa but not by any sa with fewer than $2^n - 1$ states.

In the same paper he raised the question of how sa compare with unrestricted 2dfa. In this paper we show that the ability to change the direction of head motion in the middle of the input tape allows the automaton to be exponentially more succinct.

2. Main result

Theorem. For all n there is a language A_n which is accepted by an O(n)-state 2dfa but not by any sa with fewer than $2^n - 1$ states.

The following lemma is fundamental for proving the above theorem.

Lemma 1. Let L be a language on a finite alphabet Σ such that:

(1) If w belongs to L then all substrings of w belong to L.

(2) In L there is at least one string x such that for all strings u and v in L, uxv belongs to L.

(3) There exists a string d over Σ such that:

(i) d has length 2^n,

(ii) d does not belong to L,

(iii) after the removal of any of its non empty substrings d belongs to L.

Then L cannot be accepted by an sa with fewer than $2^n - 1$ states.

* This research has been supported by a fellowship from Consiglio Nazionale delle Ricerche, Italy and in part by NSF Grant MCS-79-03767.

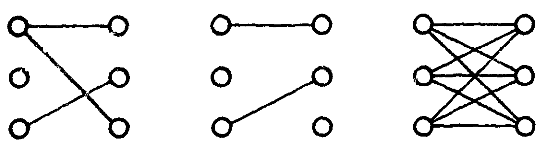
Proof. The proof can be carried out by following, step by step, the demonstration given in [2] that B_n cannot be accepted by an sa with less than $2^n - 1$ states. In fact properties (1)–(3) hold for B_n and they are the only features of B_n involved in that proof.

In order to prove the theorem we define the language A_n. Say that a bipartite graph is of type 1 if no two edges meet at a right node and of type 2 if it is a complete bipartite graph. The alphabet Σ_n of A_n consists of all bipartite graphs satisfying these two properties:

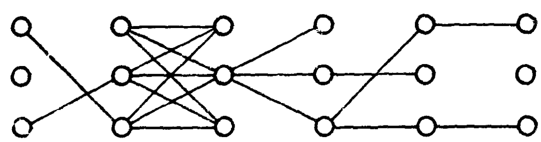
(i) The graphs have n left nodes and n right nodes.

(ii) The graphs are either of type 1 or of type 2.

A sequence of such symbols constitutes a string by identifying right and left nodes of adjacent bipartite graphs:



some members of Σ_3



a string over Σ_3

A string s over Σ_n is a member of A_n iff

(a) there is a path leading from a leftmost node of s to a rightmost one.

Definition. A chain of a string s is a maximal substring of s consisting of symbols of type 1.

Say a chain is good iff there is a path from one of its leftmost nodes to one of its rightmost nodes. Then s belongs to A_n iff

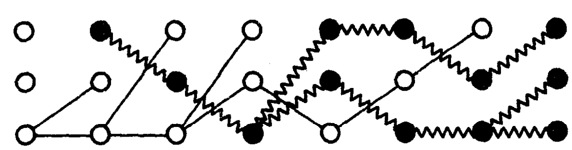
(b) all chains of s are good.

Note that a chain is circuit free and thus (b) can be checked using depth first search by a 2dfa with $O(n^2)$ states. With care one can prove that (b) can be checked by a 2dfa with 2n states. We summarize these observations in the following lemma.

Lemma 2. A_n is accepted by an O(n)-state 2dfa.

Lemma 3. A_n cannot be accepted by an sa with less than $2^n - 1$ states.

Proof. We show that A_n satisfies the conditions of Lemma 1. Let w belong to A_n and s be a substring of w. As there is a path from a leftmost node of w to a rightmost one, such a path must also connect a leftmost node of s with a rightmost node of s. Thus property (1) holds for A_n. The word consisting of a single complete bipartite graph is a valid x for property (2). Let's now construct d_n (a valid d for A_n): write down 2^n columns of n nodes numbered 1 through n (top-down). Order the subsets of $I_n = \{1, ..., n\}$ first by cardinality and then lexicographically. In the i^{th} column from the left, mark the nodes that correspond to the i^{th} subset of I_n. For $i = 1$ to $2^n - 1$, connect the 1^{st} marked node of column i with the 1^{st} marked node of column $i + 1$, 2^{nd} with 2^{nd} and so on, and let the last marked node connect to all remaining marked nodes (at most one) of column $i + 1$. For unmarked nodes, connect the last of them in column i with all unmarked nodes of column $i + 1$.



The string d_3

d_n has length 2^n. In d_n no path runs from a leftmost node to a rightmost one but the removal of any non empty substring will create one. Thus d_n has all the properties required in (3); this completes the proof.

Acknowledgment

I am sincerely grateful to David Lichtenstein for having suggested the problem and for all the generous help he gave me. Many thanks are also due to Manuel Blum, Mike Sipser, Shafi Goldwasser, Vijay Vazirani and Mike Harrison for their useful comments.

References

[1] W.J. Sakoda and M. Sipser, Nondeterminism and the size of two-way finite automata, Tenth Annual STOC (1978) 275–286.

[2] M. Sipser, Lower bounds on the size of sweeping automata, Proc. 11th Annual ACM Symposium on Theory of Computing (1979) 360–364.
This work also appeared in:
[3] S. Micali, Two way deterministic finite automata are exponentially more succinct than sweeping automata, University of California Memo No. UCB/ERL/M80/4 (1980).
and independently in:
[4] P. Berman, A note on sweeping automata, Proc. 7th Internat. Colloquium on Automata, Languages and Programming (1980) 91–97.