ELSEVIER

Information Processing Letters 59 (1996) 75–77

Information Processing Letters

A lower bound technique for the size of nondeterministic finite automata

Ian Glaister [1], Jeffrey Shallit *

Department of Computer Science, University of Waterloo, Waterloo, ON, Canada N2L 3G1

Received 13 December 1995; revised 30 May 1996
Communicated by L.A. Hemaspaandra

Abstract

In this note, we prove a simple theorem that provides a lower bound on the size of nondeterministic finite automata which accept a given regular language.

Keywords: Formal languages; Nondeterministic finite automata; Lower bound

We measure the size of an automaton by counting the number of states it contains. Given a regular language L, the well-known Myhill–Nerode theorem (e.g., [4, Theorem 3.9]) provides an efficient way to determine the smallest deterministic finite automaton (DFA) that accepts L. The smallest DFA for a given language is unique, up to the naming of the states.

Unfortunately, no such general method is known for the case of nondeterministic finite automata (NFAs). For one thing, the smallest NFA is not necessarily unique; for an example, see [1] or [5, Fig. 3, p. 167]. Furthermore, it is unlikely any such general method will be tractably computable, since it is known [6, Theorem 3.2] that the following decision problem is PSPACE-complete:

Instance: A DFA M and an integer k.
Question: Is there an NFA with $\leqslant k$ states accepting $L(M)$?

As Jiang, McDowell and Ravikumar remark [5]:

> While the standard argument based on the Myhill–Nerode equivalence relation R_L yields good lower bounds on the size of DFAs, no such methods are known for proving lower bounds on the size of NFAs.

In this note we prove a remarkably simple theorem, based on communication complexity, that gives such a lower bound. Although the lower bound provided by our theorem is not always tight, it gives good results in many cases. We emphasize that the goal of this note is *not* to provide techniques for actually finding a nondeterministic automaton of minimum size; for this problem, see, for example, [7,8,1,9].

We assume the reader is familiar with the standard notation for language theory, as provided in [4].

Theorem 1. *Let $L \subseteq \Sigma^*$ be a regular language, and*

* Corresponding author. Email: shallit@graceland.uwaterloo.ca. Research supported in part by a grant from NSERC. Please direct all correspondence to this author.

[1] Email: ian@array.ca. Current address: Array Systems Computing Inc., 1120 Finch Avenue West, 8th Floor, North York, ON, Canada M3J 3H7.

PII S0020-0190(96)00095-6

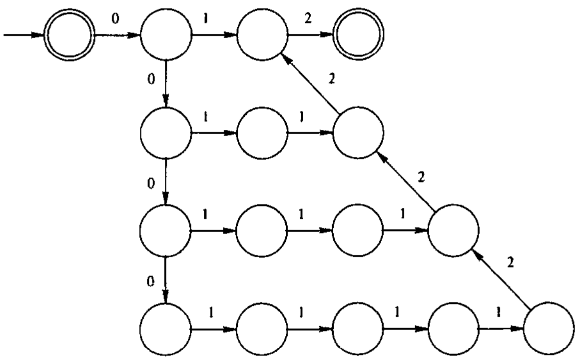

Fig. 1. An NFA accepting L_5.

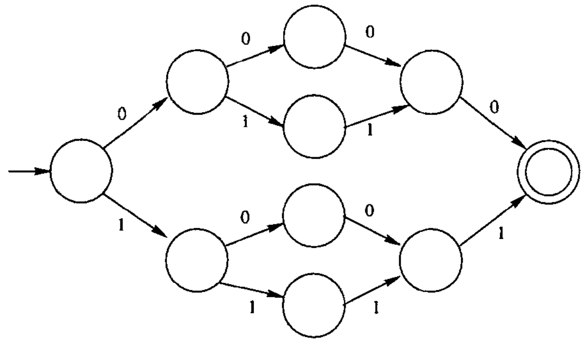

Fig. 2. An NFA accepting A_4.

suppose there exists a set of pairs $P = \{(x_i, w_i) \mid 1 \leqslant i \leqslant n\}$ *such that*

(a) $x_i w_i \in L$ *for* $1 \leqslant i \leqslant n$;

(b) $x_j w_i \notin L$ *for* $1 \leqslant i, j \leqslant n$, *and* $i \neq j$.

Then any NFA accepting L has at least n states.

Proof. Let $M = (Q, \Sigma, \delta, q_0, F)$ be any NFA accepting L, and consider the set of states $S = \delta(q_0, x_i)$. Since $x_i w_i \in L$, there must be a state $p_i \in S$ such that $\delta(p_i, w_i) \cap F$ is nonempty. In other words, there exists a state $r_i \in F$ with $r_i \in \delta(p_i, w_i)$. We claim $p_i \notin \delta(q_0, x_j)$ for all $j \neq i$. For if $p_i \in \delta(q_0, x_j)$, then $r_i \in \delta(p_i, w_i) \subseteq \delta(q_0, x_j w_i)$, so $x_j w_i \in L$, a contradiction. It follows that each set $\delta(q_0, x_i)$ contains a state p_i which is not contained in any other set $\delta(q_0, x_j)$ with $j \neq i$. Hence M has at least n states. □

In applying this theorem to any particular language L, it is of course necessary to choose the pairs (x_i, w_i) appropriately. We do not know an infallible algorithm for optimally making these choices, but the following heuristic seems to work well. Construct an NFA accepting L, and for each state q in this NFA let x_q be the shortest string such that $\delta(q_0, x_q) = q$, and let w_q be the shortest string such that $\delta(q, w_q) \in F$. Then choose the set P to be some appropriate subset of the pairs $\{(x_q, w_q) \mid q \in Q\}$.

We now give three examples of the application of this theorem.

Example 2. Let $L_k = \{0^i 1^i 2^i \mid 0 \leqslant i < k\}$. In Theorem 1 we can take as our set of pairs $P = \{(0^i 1^j, 1^{i-j} 2^i) \mid 0 \leqslant j \leqslant i < k\}$. Let $(x, w) = (0^i 1^j, 1^{i-j} 2^i)$ and $(x', w') = (0^{i'} 1^{j'}, 1^{i'-j'} 2^{i'})$ be two such distinct pairs. Then clearly $xw \in L$, but $xw' = 0^i 1^{i'+j-j'} 2^{i'}$ cannot be in L unless $i = i'$ and $j = j'$. It follows that there are at least $|P| = k(k+1)/2$ states in any NFA that accepts L_k. In fact, L_k can be accepted by an NFA with $k(k+1)/2 + 1$ states. Rather than give a formal proof, we illustrate the construction for $k = 5$ in Fig. 1.

Example 3. Let w^{R} denote the reverse of the string w, and consider the language

$$A_k = \{w \in (0+1)^k \mid w = w^{\mathrm{R}}\}$$

of palindromes of length k over a binary alphabet. In Theorem 1 we may take

$$P = \{(x, 0^{k-2|x|} x^{\mathrm{R}}) \mid |x| \leqslant k/2\} \cup \{(x 0^{k-2|x|}, x^{\mathrm{R}}) \mid |x| \leqslant (k-1)/2\}.$$

It follows that the smallest NFA accepting A_k has at least $2^{\lfloor k/2 \rfloor + 1} + 2^{\lfloor (k+1)/2 \rfloor} - 2$ states. In fact, this bound is tight, as can be easily proved by actually constructing an NFA with the given number of states that accepts A_k. Rather than give a formal proof, we illustrate the construction for $k = 4$ in Fig. 2.

While Theorem 1 is often useful for obtaining lower bounds (see [2,3]), the lower bound provided is not always tight. In fact, the lower bound provided by Theorem 1 may be arbitrarily bad compared to the true bound. Consider the following example.

Example 4. Define

$$H_k = \overline{(0^k)^+}.$$

The reader can easily verify that the hypothesis of Theorem 1 cannot be fulfilled for this language if $n > 2$. However, the smallest NFA for H_k must have at least $\log_2(k+1)$ states. To see this, observe that the smallest DFA accepting any regular language $L \neq \Sigma^*$ must have at least one more state than the length of a shortest string not in L. Hence the smallest DFA accepting H_k must have at least $k+1$ states. By the standard subset construction, the smallest NFA accepting H_k must have at least $\log_2(k+1)$ states.

References

[1] A. Arnold, A. Dicky and M. Nivat, A note about minimal non-deterministic automata, *Bull. European Assoc. Theoret. Comput. Sci.* **47** (1992) 166–169.

[2] I. Glaister and J. Shallit, Automaticity III: Polynomial automaticity, context-free languages, and fixed points of morphisms, Manuscript, 1995.

[3] I. Glaister and J. Shallit, Polynomial automaticity, context-free languages and fixed points of morphisms, in: *Proc. of MFCS '96* (1996), to appear.

[4] J.E. Hopcroft and J.D. Ullman, *Introduction to Automata Theory, Languages and Computation* (Addison-Wesley, Reading, MA, 1979).

[5] T. Jiang, E. McDowell and B. Ravikumar, The structure and complexity of minimal NFA's over a unary alphabet, *Internat. J. Found. Comput. Sci.* **2** (1991) 163–182.

[6] T. Jiang and B. Ravikumar, NFA minimization problems are hard, *SIAM J. Comput.* **22** (1993) 1117–1141.

[7] T. Kameda and P. Weiner, On the state minimization of nondeterministic finite automata, *IEEE Trans. Comput.* **19** (1970) 617–627.

[8] J. Kim, State minimization of nondeterministic machines, Tech. Rept. RC 4896, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, 1974.

[9] O. Matz and A. Potthoff, Computing small nondeterministic finite automata, in: U.H. Engberg, K.G. Larsen and A. Skou, eds., *Proc. of the Workshop on Tools and Algorithms for the Construction and Analysis of Systems*, BRICS Notes Series (May 1995) 74–88.