ELSEVIER

Information Processing Letters 58 (1996) 185–188

Information Processing Letters

The state complexity of $\overline{\Sigma^* \overline{L}}$ and its connection with temporal logic

Jean-Camille Birget [1]

Department of Computer Science and Engineering, University of Nebraska, Ferguson Hall, Lincoln, NE 68588-0115, USA

Received 25 October 1995, revised 8 February 1996
Communicated by L. Boasson

Keywords: Formal languages; Finite automata; State complexity; Temporal logic

This note answers the following question of Jean-Eric Pin. Let Σ be a finite alphabet and let $L \subseteq \Sigma^*$ be a regular language, recognized by an NFA (non-deterministic finite automaton) or a DFA (deterministic finite automaton) with n states. How many states are sufficient (and necessary in the worst case) for an NFA, respectively a DFA, if it is to recognize $\overline{\Sigma^* \cdot \overline{L}} = \Sigma^* - \Sigma^* \cdot \overline{L}$? (In general, $\Sigma^* - X = \overline{X}$ denotes the complement of a set X in Σ^*, and $X \cdot Y$ or XY denotes concatenation.) We show an upper bound of 2^{n-1} states for a complete DFA recognizing $\overline{\Sigma^* \overline{L}}$, if L has an n-state DFA. We also show that this upper bound is optimal, even if NFAs are used to recognize $\overline{\Sigma^* \overline{L}}$. If L has an n-state NFA then $\overline{\Sigma^* \overline{L}}$ has an NFA with $\leqslant 2^{n+1} + 1$ states, and this bound is close to optimal.

In spite of its complicated appearance $\overline{\Sigma^* \overline{L}}$ has a rather simple description:

$$\overline{\Sigma^* \overline{L}} = \{ w \in \Sigma^* \mid \text{every suffix of } w \text{ belongs to } L \}.$$

(Recall that the empty word and w itself are also suffixes of w.)

Note that this expression implies that $\overline{\Sigma^* \overline{L}} = \emptyset$ if L does not contain the empty word.

Connection with Temporal Logic. The motivation of Pin's question comes from the word model of Propositional Temporal Logic; for terminology and further references see [5]. Here the set of all models of a formula φ (over a fixed alphabet Σ) is a formal language $L(\varphi) \subseteq \Sigma^*$, which has the non-trivial property of being regular and aperiodic. Some of the temporal operators used in this logic are $\bigcirc$ ("*next*") and $\Diamond$ ("*eventually*", or "at some moment in the future"); there are also the usual boolean operations $^-$, $\wedge$, $\vee$. A natural dual to the "eventually" operator is the "*forever*" (or, "always in the future") operator $\Box$, defined to be $^-\Diamond^-$ ("not eventually not"). If only $\bigcirc$, $\Diamond$ (or $\Box$), and the boolean operations are used, one obtains the Restricted Propositional Temporal Logic (RPTL). One of the main results in [5] is that a language $L \subseteq \Sigma^*$ is the set of models of a formula in RPTL if and only if the syntactic semigroup of L is "locally $\mathscr{L}$-trivial" (see

[1] Supported in part by NSF grant DMS-9203981. Email: birget@cse.unl.edu.

PII S0020-0190(96)00044-0

[5] for the definition). Formulas and their models are related as follows (as is easy to check):

$$L(\overline{\varphi}) = \overline{L(\varphi)}, \qquad L(\varphi \wedge \psi) = L(\varphi) \cap L(\psi),$$
$$L(\varphi \vee \psi) = L(\varphi) \cup L(\psi), \qquad L(\bigcirc\varphi) = \Sigma \cdot L(\varphi),$$
$$L(\Diamond)\varphi) = \Sigma^* \cdot L(\varphi).$$

Thus $L(\square\,\varphi) = L(\overline{\Diamond\overline{\varphi}}) = \overline{\Sigma^* \cdot \overline{L(\varphi)}}$. In other words, in this paper we study the state-complexity of the "forever" operator.

For more information on NFAs and complete DFAs, see [6]; a DFA is "complete" if the next state is always defined. We will also use AFAs (alternating finite automata), because of their obvious ties to Logic (see [4,3,8,9] for the definition of AFA; we will follow [3]).

Theorem 1 (Upper bounds). (a) *If $L \subseteq \Sigma^*$ is recognized by an* AFA (*or, in particular, by an* NFA *or a* DFA) *with n states, then $\overline{\Sigma^*\overline{L}}$ is recognized by an* AFA *with $\leqslant n+1$ states, and $(\Sigma^*\overline{L})^{\text{rev}}$ is recognized by a* DFA *with $\leqslant 2^{n+1}$ states. Hence $\overline{\Sigma^*\overline{L}}$ is recognized by an* NFA *with $\leqslant 2^{n+1}+1$ states.*

(b) *If L is recognized by a* DFA (*complete or not*) *with n states, then $\overline{\Sigma^*\overline{L}}$ is recognized by a complete* DFA *with $\leqslant 2^{n-1}$ states.*

Theorem 2 (Lower bounds). (a) *For every $n \geqslant 2$ there exists a 3-letter alphabet Σ and a language L ($\subseteq \Sigma^*$) which is recognized by a complete* DFA *with n states, but such that every* NFA (*hence every* DFA) *recognizing $\overline{\Sigma^*\overline{L}}$ has at least 2^{n-1} states.*

(b) *For every $n \geqslant 2$ there exists a 2-letter alphabet Σ and a language L ($\subseteq \Sigma^*$) which is recognized by a complete* DFA *with n states, and which is expressible in RPTL (in fact, L is the complement of a finite language, so it can be expressed in RPTL without using $\Diamond$); however, every complete* DFA *recognizing $\overline{\Sigma^*\overline{L}}$ (or $\Sigma^* \cdot \overline{L}$) has at least 2^{n-1} states.*

Theorem 2 implies that for complete DFAs the upper bound 2^{n-1} of Theorem 1(b) is optimal; for NFAs, the upper bound in Theorem 1(a) is almost optimal.

Proof of Theorem 1(a). Suppose $L \subseteq \Sigma^*$ is recognized by an AFA $\mathbf{A}_1$ with n states, and with initial boolean function f_1. Then $\overline{L}$ is also recognized by an AFA $\mathbf{A}_2$ with n states and with initial boolean function f_2 (one only has to negate the initial boolean function: $f_2 = \bar{f}_1$). From this one obtains an AFA $\mathbf{A}_3$ with $n+1$ states, recognizing $\Sigma^* \cdot \overline{L}$ (one adds a new start state s and introduces the transitions $s \cdot a = \{s\} \cup \{\text{start states of } \mathbf{A}_2\}$, for each $a \in \Sigma$; the new initial boolean function is $f_3 = s \vee f_2$). Finally, we obtain an AFA $\mathbf{A}_4$ recognizing $\overline{\Sigma^*\overline{L}}$ by negating the initial boolean function of $\mathbf{A}_3$: $f_4 = \overline{s \vee f_2}$; the number of states of $\mathbf{A}_4$ is $n+1$.

We obtain an NFA with $2^{n+1}+1$ states for $\overline{\Sigma^*\overline{L}}$ by applying the following theorem of Kozen (see [7,4]) to the AFA $\mathbf{A}_4$: If a language R is recognized by an AFA with m states, then R^{rev} (the reverse of R) is recognized by a complete DFA with 2^m states.

Thus $(\overline{\Sigma^*\overline{L}})^{\text{rev}}$ has a complete DFA with 2^{n+1} states. By reversing this DFA (i.e., reversing the direction of every arrow, and exchanging accept and start states) we obtain an NFA with $2^{n+1}+1$ states, recognizing $\overline{\Sigma^*\overline{L}}$. (An additional state had to be added to the NFA since the DFA could have had many accept states, which would yield an NFA with many start states; but we want an NFA to have only one start state; this is a classical construction.) $\square$

Proof of Theorem 1(b). Let $\mathbf{A} = (Q, \Sigma, \cdot, q_0, F)$ be a DFA recognizing L with $|Q| = n$. Recall that $\overline{\Sigma^*\overline{L}} = \{w \in \Sigma^* \mid \text{every suffix of } w \text{ belongs to } L\}$. Since $\overline{\Sigma^*\overline{L}} = \emptyset$ if L does not contain the empty word, the claimed upper bound certainly holds in this case. Let us henceforth assume that $q_0 \in F$. The following complete DFA, inspired from the subset construction (see [6]), recognizes $\overline{\Sigma^*\overline{L}}$:

$$\mathbf{B} = \big(\{P \in \mathscr{P}(Q) \mid q_0 \in P\},\ \Sigma,\ \circ,\ \{q_0\},$$
$$\{P \in \mathscr{P}(Q) \mid q_0 \in P \text{ and } P \subseteq F\}\big);$$

here $\mathscr{P}(Q)$ denotes the power set of Q. The next-state function $\circ$ is defined as follows for $a \in \Sigma$:

$$P \circ a = \{q_0\} \cup P \cdot a = \{q_0\} \cup \{p \cdot a \mid p \in P\}.$$

Proof that $\mathbf{B}$ recognizes $\overline{\Sigma^*\overline{L}}$: $\mathbf{B}$ accepts $w = a_1 a_2 \dots a_m$ if and only if $\{q_0\} \circ a_1 a_2 \dots a_m = \{q_0\} \cup \{q_0 \cdot a_k \dots a_{m-1} a_m \mid k = 1, \dots, m\} \subseteq F$; this holds if and only if for all $k \in \{1, \dots, m\}$: $q_0 \cdot a_k \dots a_{m-1} a_m \in F$ (we already assumed $q_0 \in F$); this holds if and only if every suffix $a_k \dots a_{m-1} a_m$ of w (and the empty suffix as well, by assumption) belongs to L; this holds if and only if $w \in \overline{\Sigma^* \overline{L}}$. □

Proof of Theorem 2(a). For every $n \geqslant 1$, let $\mathbf{n} = \{1, \ldots, n\}$, and let F_n be the set of all total functions from $\mathbf{n}$ to $\mathbf{n}$. For $x \in \mathbf{n}$ and $f \in F_n$ we denote the image of x under f by $(x)f$; in this notation, functions compose from left to right, e.g., $(x)(f_1 f_2 f_3) = (((x)f_1)f_2)f_3$.

We will pick F_n as our alphabet, and for $n \geqslant 2$ we consider the following language:

$$L_n = \{w \in (F_n)^* \mid (1)f_1 \ldots f_k \neq 2, \text{ where } w = (f_1, \ldots, f_k),\ k \geqslant 0\}.$$

(The empty word is also in L_n, when $k = 0$ in the above definition.)

Then L_n is recognized by the complete DFA $\mathbf{A} = (\mathbf{n}, F_n, \cdot, 1, \mathbf{n} - \{2\})$, where the next-state function "$\cdot$" is defined by $i \cdot f = (i)f$, for $i \in \mathbf{n}$ and $f \in F_n$. So L_n has an n-state complete DFA.

The alphabet F_n has size n^n, but we shall see later how one can modify the above example (without changing the main properties of the languages) so that the alphabet has size 3.

Fact 1. *The minimum complete* DFA **B** *of* $\overline{\Sigma^* \overline{L_n}}$ *has* 2^{n-1} *states.*

Proof. We consider the complete DFA **B** that was constructed in the proof of Theorem 1(b), and we show that **B** is minimum for this example. Thus the minimum complete DFA for $\overline{\Sigma^* \overline{L_n}}$ has 2^{n-1} states.

Here $\mathbf{B} = (\{P \subseteq \mathbf{n} \mid 1 \in P\}, F_n, \circ, \{1\}, \{P \subseteq \mathbf{n} \mid 1 \in P \text{ and } 2 \notin P\})$, where the next-state function $\circ$ is given by $P \circ a = \{1\} \cup \{(i)a \mid i \in P\}$ when $a \in F_n$ and $P \subseteq \mathbf{n}$. Let us prove minimality of **B**.

Claim 1 (Reachability from the start state $\{1\}$). For every $P \subseteq \mathbf{n}$ with $1 \in P$ there exists $u_P \in (F_n)^*$ such that $\{1\} \circ u_P = P$.

Proof of Claim 1. Let $P = \{1, p_1, \ldots, p_k\} \subseteq \mathbf{n}$ with $1 < p_1 < \cdots < p_k$. We let $u_P = f_1 f_2 \ldots f_k \in (F_n)^*$, where f_i (for $1 \leqslant i \leqslant k$) is defined by: $(1)f_i = p_i$, and $(x)f_i = x$ for $x \neq 1$. It is straightforward to check that $\{1\} \circ f_1 = \{1, p_1\}$, $\{1, p_1\} \circ f_2 = \{1, p_2, p_1\}$, $\{1, p_2, p_1\} \circ f_3 = \{1, p_3, p_2, p_1\}$, etc., and $\{1\} \circ u_P = P$.

Claim 2 (Co-reachability). For every $P \subseteq \mathbf{n}$ (with $1 \in P$) there exists $w \in (F_n)^*$ such that $2 \notin P \circ w$ (i.e., $P \circ w$ is an accept state).

Proof of Claim 2. Simply pick w to be the constant function $c_1 \in F_n$ defined by $(x)c_1 = 1$ for all $x \in \mathbf{n}$. Then $P \circ c_1 = \{1\} \cup P \cdot c_1 = \{1\}$ (an accept state of **B**).

Claim 3 (Distinguishability of all the states). For every $P_1, P_2 \subseteq \mathbf{n}$ with $1 \in P_1$, $1 \in P_2$ and $P_1 \neq P_2$, there exists $w \in (F_n)^*$ such that exactly one of $P_1 \circ w$ and $P_2 \circ w$ is an accept state.

Proof of Claim 3. Since $P_1 \neq P_2$, either $P_1 - P_2 \neq \emptyset$ or $P_2 - P_1 \neq \emptyset$. Let $q \in P_1 - P_2$, if $P_1 - P_2 \neq \emptyset$ (if $P_2 - P_1 \neq \emptyset$ the proof is similar). Let w be the function $f \in F_n$ defined by $(q)f = 2$, and $(x)f = 1$ for $x \neq q$. Then $P_1 \circ w = \{1, 2\}$, and $P_2 \circ w = \{1\}$, so $P_1 \circ w$ is not an accept state but $P_2 \circ w$ is an accept state.

This completes the proof of Fact 1. □

Fact 2. *Every* NFA *recognizing* $\overline{\Sigma^* \overline{L_n}}$ *has* $\geqslant 2^{n-1}$ *states.*

The following lemma from [1,2] is a convenient tool for proving lower bounds on the number of states of NFAs. (See [1] for a proof.)

Lemma. *Let* $R \subseteq \Sigma^*$ *be a regular language, and let* X *be a finite set. Assume that with every* $x \in X$ *one can associate words* u_x *and* $v_x \in \Sigma^*$ *such that*
(1) $(\forall x \in X)\ u_x v_x \in R$,
(2) $(\forall x, y \in X$ *with* $x \neq y)$ $u_x v_y \notin R$ **or** $u_y v_x \notin R$.
Then every NFA *recognizing* R *has* $\geqslant |X|$ *states.*

Proof of Fact 2. We apply the lemma. For X we take the set $X = \{P \subseteq \mathbf{n} \mid 1 \in P\}$. Then $|X| = 2^{n-1}$. With every $P \in X$ we associate two words $u_P, v_P \in (F_n)^*$ as follows: u_P is the word defined in the proof of Fact 1, Claim 1 (Reachability from $\{1\}$); and v_P is the function in F_n defined as follows (for any q): $(q)v_P = 1$ if $q \in P$, and $(q)v_P = 2$ if $q \notin P$ (so v_P is just a one-letter word.)

Then we have:

(1) $u_P v_P \in \overline{\Sigma^* \overline{L_n}}$: Indeed, $\{1\} \circ u_P v_P = P \circ v_P$, by the proof of Claim 1. Moreover, $P \circ v_P = \{1\}$, so $u_P v_P$ is accepted by the DFA **B** of $\overline{\Sigma^* \overline{L_n}}$.

(2) $u_P v_S \notin \overline{\Sigma^* \bar{L}_n}$ or $u_S v_P \notin \overline{\Sigma^* \bar{L}_n}$ if $P \neq S$: Indeed, if $P - S \neq \emptyset$ then $\{1\} \circ u_P v_P = P \circ v_S = \{1, 2\}$ (which is a non-accept state of **B**, as it contains 2), so $u_P v_S \in \Sigma^* \bar{L}_n$ similarly, if $S - P \neq \emptyset$ then $u_S v_P \in \Sigma^* \bar{L}_n$.

This proves Fact 2. □

Reducing the alphabet size to 3. It is a classical fact from semigroup theory that the set F_n of all total functions from **n** to **n** is generated, under functional composition, by just three functions. As generators one can use the three functions α, β, γ defined as follows:

- $(x)\alpha = x + 1 \bmod n$ for all $x \in \mathbf{n}$;
- $(1)\beta = 2$, $(2)\beta = 1$, and $(x)\beta = x$ for $3 \leqslant x \leqslant n$;
- $(2)\gamma = 1$, and $(x)\gamma = x$ for $2 \leqslant x \leqslant n$.

Next, we replace the alphabet F_n by $\{\alpha, \beta, \gamma\}$, and we replace L_n by the language $L_n \cap \{\alpha, \beta, \gamma\}^*$. Since $\{\alpha, \beta, \gamma\}$ generates F_n one can check that all the properties that we proved about L_n still hold. □

Proof of Theorem 2(b). We let $\bar{L}_n$ be the finite language $a\{a, b\}^{n-2}$, for $n \geqslant 2$. The alphabet is $\Sigma = \{a, b\}$. Then $\Sigma^* \bar{L}_n = \Sigma^* a \Sigma^{n-2}$. It is well known that the minimum complete DFA for this language (as well as for the complement) has exactly 2^{n-1} states (as observed by Paterson, quoted in [10]).

Note that for $\bar{L}_n = a\{a, b\}^{n-2}$, the languages $\Sigma^* \bar{L}_n$ and $\overline{\Sigma^* \bar{L}_n}$ are both accepted by NFAs with n states (for $\Sigma^* \bar{L}_n$ this is a well-known exercise, see [6]; for its complement, observe that $\overline{\Sigma^* \bar{L}_n} = \Sigma^{\leqslant n-2} \cup \Sigma^* b \Sigma^{n-2}$, which directly yields an NFA with $2n - 2$ states; many of these states can be identified in pairs). □

References

[1] J.C. Birget, Intersection and union of regular languages, and state-complexity, *Inform. Process. Lett.* **43** (1992) 185–190.

[2] J.C. Birget, Partial orders on words, minimal elements of regular languages, and state-complexity, *Theoret. Comput. Sci.* **119** (1993) 267–291.

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

[4] A. Chandra, D. Kozen and L. Stockmeyer, Alternation, *J. ACM* **28** (1981) 114–133.

[5] J. Cohen, D. Perrin and J.-E. Pin, On the expressive power of temporal logic, *J. Comput. System Sci.* **46** (1993) 271–294.

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

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

[8] E. Leiss, Succinct representation of regular languages by boolean automata, *Theoret. Comput. Sci.* **13** (1981) 323–330.

[9] E. Leiss, Succinct representation of regular languages by boolean automata, Part II, *Theoret. Comput. Sci.* **38** (1985) 133–136.

[10] A.R. Meyer and M.J. Fischer, Economy of description by automata, grammars, and formal systems, in: *Proc. 12th IEEE Ann. Symp. on Switching and Automata Theory* (1971) 188–191.