Intersection 1992

Information Processing Letters 43 (1992) 185–190
North-Holland

28 September 1992

Intersection and union of regular languages and state complexity

Jean-Camille Birget

Department of Computer Science and Engineering, University of Nebraska, Lincoln, NE 68588, USA

Communicated by D. Dolev
Received 2 August 1991
Revised 13 July 1992

Keywords: Formal languages; finite automata; state complexity

A natural complexity measure for regular languages is the *state complexity*, i.e., the number of states needed by various finite-state devices (deterministic, non-deterministic, or other finite automata) in order to recognize the language. Often the state complexity influences the computational complexity of algorithms that use regular languages. Among the most fundamental operations that preserve regularity are intersection and union. It is well known that if k languages are each recognized by non-deterministic (or deterministic) finite automata with n states, then their intersection is recognized by a non-deterministic (respectively deterministic) finite automaton with n^k states (obtained as the cartesian product of the k given automata; see [3]). Until recently little was known about lower-bounds or, in particular, about the optimality of the upper-bounds. In [6] Ravikumar considers the following problem. Given n languages $L_n^{(0)}, L_n^{(1)}, \ldots, L_n^{(n-1)}$, each of which is recognized by a DFA (deterministic finite automaton) with n states: what is the number of states of the minimum DFA for the intersection $\bigcap_i L_n^{(i)}$ in the worst case? He proves a lower bound of $n! + 1$; but, as Leiss [5] points out, this does not completely solve the problem, since the best known upper bound is n^n. Below we show a lower bound of n^n; so the well-known upper bound is actually optimal. We extend this result to non-deterministic finite automata. We also consider the operation of union of regular languages.

Unless the contrary is stated, the DFA's that we will consider are "complete" (i.e., for every state and every input there is a next state). For notation and the definition of DFA (deterministic finite automaton) and NFA (non-deterministic finite automaton) we will follow [3]. A partial DFA is an NFA whose next-state relation is a partial function (i.e., for every state and every input letter there is one or no next state). An AFA (alternating finite automaton, see [2] and [4]) is a generalization of an NFA; an AFA has a state graph like an NFA but, in addition, every state has a boolean function of states attached to it; acceptance of an input is decided by these boolean functions (in the case of an NFA these boolean functions are the OR of some states); for the exact definition (which is rather long) see [2] or [4]. In this paper AFA's are only mentioned in passing; the reader can skip any mention of AFA's without loss of continuity.

Correspondence to: J.-C. Birget, Department of Computer Science and Engineering, University of Nebraska, Ferguson Hall, Lincoln, NE 68588-0115, USA. Email: birget@cse.unl.edu.

Remark 8. The *union* of k ($\leqslant n$) languages (each of which has an n-state DFA) needs a DFA with $\geqslant n^k$ states; this follows from Example 5, de Morgan's law, and the fact that the minimum complete DFA's for a language, respectively its complement, have the same number of states (one just exchanges accept and non-accept states).

Proof of the Corollary (Alternating automata)

We use the theorem above (and Remark 7), and the following theorem of Leiss, Brzozowski, and Kozen [4]: A language L has an AFA with $\leqslant n$ states iff L^{rev} (the reverse of L) has a DFA with $\leqslant 2^n$ states. Here we use AFA's in their full generality (as in [4]). Now we just choose $A_n^{(i)}$ to be $(L_{2n}^{(i)})^{\text{rev}}$, where $L_{2n}^{(i)}$ is the language introduced in Example 5 (but with n replaced by $2n$). Remark 8 applies again for the union.

Proof of Theorem 2 (Non-determinism)

The next lemma describes a general lower-bound technique (from [1]). We then introduce three examples. Example 9 gives n languages each of which is recognized by a NFA with n states, but every NFA for the intersection needs n^n states; the alphabet has size 4. Example 10 gives n languages each of which is recognized by a DFA with n states, but every NFA for the intersection needs $n!$ states; the alphabet has size 3. Example 11 gives n languages each of which is recognized by a DFA with n states, but every NFA for the intersection needs n^n states; the alphabet has size $O(n)$. Remark 7 applies again when we consider the intersection of k, rather than n, languages. Union is dealt with in Example 12. We will make crucial use of the following important lemma.

Lemma (Lower-bound argument for the state-complexity of NFA's). *Let $L \subseteq \Sigma^*$ be a regular language, and let X be a finite set. Suppose that to every $x \in X$ we can associate two words $u_x, v_x \in \Sigma^*$ in such a way that*: (1) *For all $x \in X$: $u_x v_x \in L$.* (2) *For all $x, y \in X$ with $x \neq y$: $u_x v_y \notin L$ or $u_y v_x \notin L$. Then, every NFA recognizing L must have $\geqslant |X|$ states.*

Proof. Let $B = (Q, \Sigma, \delta, q_0, F)$ be an NFA recognizing L. To every $x \in X$ we associate a state $q_x \in Q$ (i.e., we choose a function $x \in X \to q_x \in Q$) in such a way that $q_x \in \delta(q_0, u_x)$ and $\delta(q_x, v_x)$ contains some state in F (i.e., q_x occurs on an accepting path on input $u_x v_x$, and is reached after u_x was read; since $u_x v_x \in L$, q_x exists). We have then:

Claim. If $x \neq y$ then $q_x \neq q_y$ (from this it follows that $|Q| \geqslant |X|$).

Proof. If (by contradiction) $x \neq y$ but $q_x = q_y$ then $u_x v_y$ and $u_y v_x$ would both be accepted by B (for $u_x v_y$ we start in state q_0, read u_x and reach state q_x; then from state $q_x = q_y$ we can reach an accept state by reading v_y; for $u_y v_x$ the argument is similar). This proves the lemma. □

Example 9. Let $F_n^{-1} = \{f^{-1} \mid f \in F_n\}$, where f^{-1} denotes the inverse of f (usually f^{-1} is not a function, just a relation). We still use $\cdot$ to denote the composition of relations; for a relation $R \subseteq \boldsymbol{n} \times \boldsymbol{n}$, and $x \in \boldsymbol{n}$, we define $(x)R = \{y \in \boldsymbol{n} \mid (x, y) \in R\}$. Let $L_n^{(i)} = \{G_1 G_2 \cdots G_m \in (F_n \cup F_n^{-1})^* \mid m \in \mathbb{N}, (i, i) \in G_1 \cdot G_2 \cdot \ldots \cdot G_m\}$; this language is recognized by the NFA with n states $(\boldsymbol{n}, F_n \cup F_n^{-1}, \delta, i, \{i\})$, where δ is defined by $\delta(q, G) = (q)G$, for all $q \in \boldsymbol{n}$ and $G \in F_n \cup F_n^{-1}$. Then $L_n = \bigcap_i L_n^{(i)}$ is the language $\{G_1 G_2 \cdots G_m \in (F_n \cup F_n^{-1})^* \mid m \in \mathbb{N}, G_1 \cdot G_2 \cdot \ldots \cdot G_m$ is a *reflexive* relation$\}$.

Claim. Any NFA recognizing $L_n = \bigcap_i L_n^{(i)}$ needs $\geqslant n^n$ states.

Proof. We apply the lemma: pick $X = F_n$, and for every $f \in F_n$, let $u_f = f$ and $v_f = f^{-1}$. Then $f \cdot f^{-1}$ is a reflexive relation; on the other hand $g \cdot f^{-1}$ is not reflexive when $f \neq g$. (Indeed: Let $k \in \boldsymbol{n}$ be such that $(k)f \neq (k)g$. Then $k \notin ((k)g)f^{-1} = (k)g \cdot f^{-1}$, by the definition of f^{-1}. Thus $(k, k) \notin g \cdot f^{-1}$.) The claim now follows from the lemma. □

To obtain a four-letter alphabet we take the three generators α, β, γ of F_n and add γ^{-1}; we do not need to add α^{-1} and β^{-1} since they

belong to the symmetric group, which is generated by α and β.

Example 10 (Example 3, of Ravikumar, revisited). We use the same languages as in Example 3, $L_n^{(i)} = \{f_1 f_2 \cdots f_m \in F_n^* \mid m \in \mathbb{N},\ (i)f_1 \cdot f_2 \cdot \ldots \cdot f_m = i\}$; but we prove a stronger result.

Claim. Any NFA recognizing $L_n = \bigcap_i L_n^{(i)}$ needs $\geqslant n!$ states.

Proof. We apply the lemma with $X =$ set of all permutations on $\boldsymbol{n}$; for each permutation p we pick $u_p = p$ and $v_p = p^{-1}$. Then $p \cdot p^{-1}$ is the identity function; on the other hand $g \cdot f^{-1}$ is not the identity function for any permutations $f \neq g$. The claim then follows from the lemma. □

To obtain a three-letter alphabet we take the three generators α, β, γ of F_n.

Example 11. The alphabet will be $F_n \cup \{a_f \mid f \in F_n\}$, where each a_f is a new letter. We define the n languages $L_n^{(i)}$ $(i = 0, 1, \ldots, n-1)$ so that $L_n^{(i)}$ is recognized by the DFA $A_i = (\boldsymbol{n}, F_n \cup \{a_f \mid f \in F_n\}, \delta_i, i, \boldsymbol{n} - \{i\})$; here for all $i \in \boldsymbol{n}$, $q \in \boldsymbol{n}$, $f \in F_n$: $\delta_i(q, f) = (q)f$, and $\delta_i((i)f, a_f) = i + 1 \pmod n$, $\delta_i(q, a_f) = 1$ if $q \neq (i)f$. Then $L_n = \bigcap_i L_n^{(i)} = \{b_1 b_2 \cdots b_m \in (F_n \cup \{a_f \mid f \in F_n\})^* \mid m \in \mathbb{N},$ and $\forall i \in \boldsymbol{n}$: $\delta_i(i, b_1 b_2 \cdots b_m) \neq i\}$.

Claim. Any NFA recognizing $L_n = \bigcap_i L_n^{(i)}$ needs n^n states.

Proof. We apply the lemma with $X = F_n$; for each $f \in F_n$ we pick $u_f = f$ and $v_f = a_f$. Then for all i we have: $\delta_i(i, fa_f) = \delta_i((i)f, a_f) = i + 1 \neq i \pmod n$; so $fa_f \in L_n$. On the other hand, if $f \neq g$ then let j be such that $(j)f \neq (j)g$; then $\delta_j(j, ga_f) = \delta_j((j)g, a_f) = j$; so $ga_f \notin L_n$. Now the claim follows from the lemma. □

Example 11′ (Example 11 with reduced alphabet). To reduce the alphabet to size $O(n)$, we consider the three generators α, β, γ of F_n, and we take the alphabet $\Sigma_n = \{\alpha, \beta, \gamma\} \cup \{\alpha, \beta, \gamma\} \times \boldsymbol{n}$. We redefine the above n automata and languages (over the new alphabet) as follows, for $i = 0, 1, \ldots, n-1$: $L_n^{(i)}$ is recognized by the DFA $A_i = (\boldsymbol{n}, \Sigma_n, \delta_i, i, \boldsymbol{n} - \{i\})$, where δ_i is defined by $\delta_i(q, (\xi, i)) = \delta_i(q, \xi) = (q)\xi$ for all $q \in \boldsymbol{n}$, $\xi \in \{\alpha, \beta, \gamma\}$, and $\delta_j(q, (\xi, i)) = q$ for all $q \in \boldsymbol{n}$, $\xi \in \{\alpha, \beta, \gamma\}$, $j \in \boldsymbol{n}$, $j \neq i$.

Claim. Any NFA recognizing $\bigcap_i L_n^{(i)}$ needs n^n states.

Proof. We apply the lemma with $X = F_n$. For each $f \in F_n$ we pick u_f to be a factorization of the function f over the set of generators $\{\alpha, \beta, \gamma\}$; so $u_f \in \{\alpha, \beta, \gamma\}^*$. For v_f we take the concatenation of n words $v_f^{(0)} \cdots v_f^{(n-1)}$, where $v_f^{(i)} \in (\{\alpha, \beta, \gamma\} \times \{i\})^*$ is obtained as follows (for $i = 0, 1, \ldots, n-1$):

We consider the function $\varphi_{f,i} : \boldsymbol{n} \to \boldsymbol{n}$ defined by $(x)\varphi_{f,i} = i$ if $x \neq (i)f$, and $(x)\varphi_{f,i} = i + 1 \bmod n$ if $x = (i)f$. Now we factor the function $\varphi_{f,i}$ over the generators $\{\alpha, \beta, \gamma\}$: $\varphi_{f,i} = \xi_1 \cdots \xi_N$ for some $N > 0$ and $\xi_1, \ldots, \xi_N \in \{\alpha, \beta, \gamma\}$. Finally, the word $v_f^{(i)}$ is defined as $v_f^{(i)} = (\xi_1, i) \cdots (\xi_N, i) \in (\{\alpha, \beta, \gamma\} \times \{i\})^*$. Note that, since $v_f^{(i)} \in (\{\alpha, \beta, \gamma\} \times \{i\})^*$, we have $\delta_j(q, v_f^{(i)}) = q$ for all $q \in \boldsymbol{n}$, $j \in \boldsymbol{n}$, $j \neq i$.

Now, for all $i \in \boldsymbol{n}$ we have: $\delta_i(i, u_f v_f) = \delta_i((i)f, v_f) = \delta_i((i)f, v_f^{(i)}) = ((i)f)\varphi_{f,i} = i + 1 \neq i \pmod n$; so $u_f v_f \in L_n$. On the other hand, if $f \neq g$ let j be such that $(j)f \neq (j)g$; then $\delta_j(j, u_g v_f) = \delta_j((j)g, v_f) = \delta_j((j)g, v_f^{(i)}) = ((j)g)\varphi_{f,j} = j$; so $u_g v_f \notin L_n$. The claim now follows from the lemma. □

Example 12 (Union). Let PF_n be the set of all *partial* functions from $\boldsymbol{n}$ to $\boldsymbol{n}$. A partial function from $\boldsymbol{n}$ to $\boldsymbol{n}$ is a function whose domain and range are (possibly different) subsets of $\boldsymbol{n}$. The semigroup PF_n can be generated, under composition, by just four generators (e.g., the three generators α, β, γ of F_n, together with the partial function ζ defined by: $(x)\zeta = x$ for $x > 0$, and $(0)\zeta$ is undefined).

For every $n \geqslant 1$, let $L_n^{(i)} = \{f_1 f_2 \cdots f_m \in PF_n^* \mid m \in \mathbb{N},\ (i)f_1 \cdot f_2 \cdot \ldots \cdot f_m = i\}$; here our alphabet is PF_n; PF_n^* is the set of all words over this alphabet. The language $L_n^{(i)}$ is recognized by the n-state *partial* DFA $A_i = (\boldsymbol{n}, PF_n, \delta_i, i, \{i\})$, where $\delta_i(q, f) = (q)f$, for all $q \in \boldsymbol{n}$, $f \in PF_n$, $i \in \boldsymbol{n}$. (By definition, a partial DFA is an NFA $(Q, \Sigma, \delta, q_0, Q_F)$ whose next-state relation δ is a partial function; so for any $q \in Q$ and $a \in \Sigma$, $\delta(q,a)$ contains one element or is empty.)

Claim. Every NFA recognizing $\bigcup_i L_n^{(i)}$ needs $\geqslant n^2 + 1$ states.

Proof. We first prove the lower bound n^2 by

applying the lemma; an additional argument will give n^2+1. We pick $X = \boldsymbol{n} \times \boldsymbol{n}$, and for every $(i, x) \in X$, let $u_{i,x}$ and $v_{i,x}$ be the partial functions defined respectively by: $(k)u_{i,x} = x$ if $k = i$ (undefined otherwise); $(k)v_{i,x} = i$ if $k = x$ (undefined otherwise). Clearly then $u_{i,x}v_{i,x} \in L_n$. But when $(i, x) \neq (j, y)$ one checks easily that $u_{i,x}v_{j,y} \notin L_n$. By the lemma it follows now that the NFA has $\geqslant n^2$ states.

To obtain the lower bound n^2+1 we show that none of the NFA states among the n^2 whose existence we have proved so far, could be the start state; so we need at least one additional state which serves as the start state. Recall from the proof on the lemma that there is a one-to-one correspondence between the set of NFA-states $\{q_{i,x} | (i, x) \in X\}$ that we have found, and the set $\{(u_{i,x}, v_{i,x}) | (i, x) \in X\}$; also in the NFA we have $q_{i,x} \in \delta(q_0, u_{i,x})$, and $\delta(q_{i,x}, v_{i,x})$ contains some accept state (where δ is the transition relation of the NFA, and q_0 is its start state). Assume, by contradiction, that some $q_{i,x}$ is the start state (i.e., $q_{i,x} = q_0$). Then $v_{i,x}$ would be accepted, so $i = x$ by the definition of L_n; moreover $q_{i,x} = q_0 \in \delta(q_0, u_{i,i})$. Also, $\delta(q_{i,x}, v_{j,j})$ contains some accept state (for all j, since $v_{j,j} \in L_n$). This would imply that $u_{i,i}v_{j,j}$ is accepted, for all j; but we saw that this is not correct when $i \neq j$. □

References

[1] J.-C. Birget, Partial orders on words, minimal elements of regular languages, and state complexity, *Theoret. Comput. Sci.*, to appear.

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

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

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

[5] E. Leiss, Some comments on a recent note of Ravikumar, *SIGACT News* **22** (1) (1991) 64.

[6] R. Ravikumar, Some applications of a technique by Sakoda and Sipser, *SIGACT News* **21** (4) (1990) 73–77.

[7] W. Sakoda and M. Sipser, Non-determinism and the size of two-way automata, in: *Proc. 10th ACM Symp. on Theory of Computing* (1978) 275–286.

[8] S. Yu and Q. Zhuang, On the state complexity of intersection of regular languages, *SIGACT News* **22** (3) (1991) 52–54.