Running Time Complexity of Printing an Acyclic Automaton

Franck Guingne[1,2], André Kempe[1], Florent Nicart[1,2]

[1] Xerox Research Centre Europe – Grenoble Laboratory
6 chemin de Maupertuis – 38240 Meylan – France
Firstname.Lastname@xrce.xerox.com – http://www.xrce.xerox.com

[2] Laboratoire d'Informatique Fondamentale et Appliquée de Rouen
Faculté des Sciences et des Techniques – Université de Rouen
76821 Mont-Saint-Aignan – France
firstname.lastname@dir.univ-rouen.fr – http://www.univ-rouen.fr/LIFAR/

Abstract. This article estimates the worst-case running time complexity for traversing and printing all successful paths of a normalized trim acyclic automaton. First, we show that the worst-case structure is a festoon with distribution of arcs on states as uniform as possible. Then, we prove that the complexity is maximum when we have a distribution of e (Napier constant) outgoing arcs per state on average, and that it can be exponential in the number of arcs.

1 Introduction

This article takes place in the scope of the study of complexity of automata algorithms (Yu, Zhuang, and Salomaa, 1994), and particularly in the study of the worst-case complexity (Nicaud, 2000). We estimate the worst-case running time complexity for traversing and printing all successful paths of a finite-state automaton. The number of states and arcs are given, and the structure is unknown. This task occurs, e.g., when all words of a natural-language lexicon represented through an automaton are printed into a file.

By "printing a path" we mean that the label of the path is written out when its final state is reached. Hence, the complexity of this part of the task depends on both the number and the length of all paths. The required traversal of paths has in general a much lower complexity because prefixes shared by several paths are traversed only once, so that many paths are not traversed in full length

We start from the worst-case structure and show that any other structure decreases the complexity of the task. We restrict our analysis to acyclic automata with a single initial state and a single final state.

The article is structured as follows: Section 2.1 recalls some basic notions concerning automata and lists the assumptions made for all following estimations. Section 3 shows the automaton structure that maximizes the analyzed complexity. Section 4 and 5 estimate the complexity for different cases, and Section 6 reports some numerical calculations, w.r.t. the number of arcs. Section 7 presents our outcomes in a concise form and concludes the article.

2 Preliminaries

2.1 Automata

According to (Eilenberg, 1974; Hopcroft, Motwani, and Ullman, 2001), an *automaton* A is defined by the 5-tuple $\langle \Sigma, Q, I, F, E \rangle$ where

Σ	is the finite alphabet
Q	is the finite set of states
$I \subseteq Q$	is the set of initial states
$F \subseteq Q$	is the set of final states
$E \subseteq Q \times \Sigma \times Q$	is the finite set of arcs

An automaton is said to be *normalized* if and only if it has exactly one initial state with no incoming arc and one final state with no outgoing arc (Berstel, 1989).

A state s is *reachable* (resp. *coreachable*) if there exists a path from some state of I to s (resp. a path from s to some state of F); an automaton is said to be *trim* if and only if all its states are reachable and coreachable (Perrin, 1990).

For any arc $e \in E$ we denote by

$p(e)$	$p : E \to Q$	the source state of e
$n(e)$	$n : E \to Q$	the target state of e

Symbols are required only for printing out the paths. They are irrelevant in the estimation of the complexity since the complexity for printing out a path does not depend on the symbols themselves.

A *path* π of length $l = |\pi|$ is a sequence of arcs $e_1 e_2 \cdots e_l$ such that $n(e_i) = p(e_{i+1})$ for all $i \in [\![1, l-1]\!]$. A path is said to be *successful* if and only if $p(e_1) \in I$ and $n(e_l) \in F$. The set of all successful paths of A is denoted by Π.

2.2 Conventions and assumptions

To simplify our notation, we will denote by:

$a = \|E\|$	the number of arcs in A
$s = \|Q\|$	the number of states in A
$p = \|\Pi\|$	the number of successful paths in A

The following analysis is made for an automaton A under the assumption that:

- A is acyclic
- A is normalized
- A is trim
- a and s are given

No more assumptions are needed for our study. Our outcomes are independent of possible additional properties of the automaton, such as determinism, ε-arcs, multiplicities (or weights).

3 Worst-Case Structure

This section introduces the structure that maximizes the complexity of traversing and printing all successful paths of a normalized acyclic automaton. We start from the worst-case structure and show that any other structure decreases the complexity.

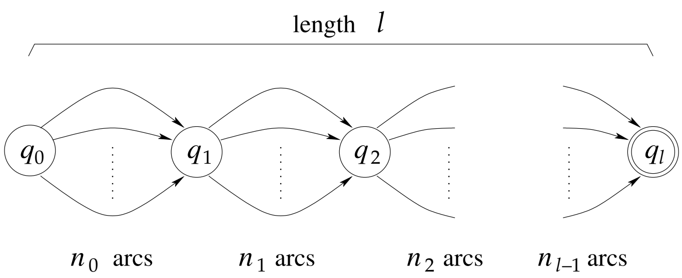

Fig. 1. Worst-Case structure of a normalized acyclic automaton.

Let A be an automaton that satisfies the assumptions in section 2.2 and has the structure shown in Figure 1, with a arcs, s states, and p paths of length l. Every state q_i, except the last (final) one has n_i outgoing arcs leading to the next state q_{i+1}.

Only a and l are fixed. According to the above structure, $s = l + 1$. We will, however, discuss alternative structures in the case of splitting a state (Figure 2) where $s > l + 1$ (s will be temporarily variable).

Since the analyzed complexity depends on both p and l, and since l is fixed at present, the maximum of the complexity is reached with the maximum of p. Hence we will maximize p in this section.

Proposition 1. *Let us consider a structure as shown in Figure 1. Let $n = \lfloor \frac{a}{l} \rfloor$, and let q_i, for $i \in [\![0, l-1]\!]$, such that*

$$out(q_i) = \begin{cases} n & if \quad i < l \text{ - } (a \bmod l) \\ n{+}1 & otherwise. \end{cases} \tag{1}$$

Then the maximum number of paths is:

$$P_{max} = \prod_{i=0}^{l-1} out(q_i) \tag{2}$$

When $a \mid l$ (l divides a) we denote

$$P_{max} = p_{\text{uni}} \; = \; n^l \tag{3}$$

With respect to the notion of hammock used in (Caron and Ziadi, 2000) and (Giammarresi, Ponty, and Wood, 2001), the structure defined by Proposition 1 is a uniform acyclic multi-hammock. In the following we call it a *festoon*.

Proof. Any of the following changes to this structure will reduce the number of paths.

1. **Moving arcs to other states:** if one arc is moved from q_i to q_j, so that q_i will have $n-1$, q_j will have $n+1$, and all the other states will have n arcs, then the number of paths will decrease to:

$$\begin{aligned} p_1 &= n^{l-2}\ (n-1)\ (n+1) = n^{l-2}\ (n^2-1) = n^l - n^{l-2} \\ &= p_{\text{uni}} - n^{l-2} \end{aligned} \tag{4}$$

If k arcs are moved in that way, the number of paths decreases as well:

$$\begin{aligned} p_k &= n^{l-2}\ (n-k)\ (n+k) = n^{l-2}\ (n^2-k^2) \\ &= p_{\text{uni}} - k^2\ n^{l-2} \end{aligned} \tag{5}$$

If uniform distribution is impossible because $\frac{a}{l} \notin I\!N$ then the maximum number of paths is reached when the distribution of the arcs is given by the function $out(q)$ i.e., n or $n+1$ arcs per state. For a length $l=l_1+l_2$ with l_1 states having n arcs each and l_2 states having $n+1$ arcs each, the number of path is:

$$P_{max} = n^{l_1} \cdot (n+1)^{l_2} \tag{6}$$

If we move an arc from an $n+1$-arcs to an n-arcs section then obviously the number of paths does not change. However, if we move an arc from an $n+1$-arcs to another $n+1$-arcs section, then the number of path decreases to:

$$\begin{aligned} p &= n^{l_1} \cdot (n+1)^{l_2-2}\ n(n+2) = n^{l_1} \cdot (n+1)^{l_2-2}\ ((n+1)^2-1) \\ &= n^{l_1} \cdot (n+1)^{l_2} - n^{l_1}\ (n+1)^{l_2-2} \\ &= P_{max} - n^{l_1}\ (n+1)^{l_2-2} \end{aligned} \tag{7}$$

and if we move an arc from an n-arcs to another n-arcs section, it decreases (symmetrically) to:

$$p = P_{max} - (n+1)^{l_2}\ n^{l_1-2} \tag{8}$$

Any other move of k arcs between any two sections can be obtained by combining the listed moves.

2. **Splitting of states:** If there are ϱ prefixes ending and σ suffixes starting in a state q then the number of paths traversing q is (Figure 2a):

$$\dot{p} = \varrho \cdot \sigma \tag{9}$$

If we split q and its sets of prefixes and suffixes, so that there will be two new states, q_1 with ϱ_1 prefixes and σ_1 suffixes, and q_2 with ϱ_2 prefixes and

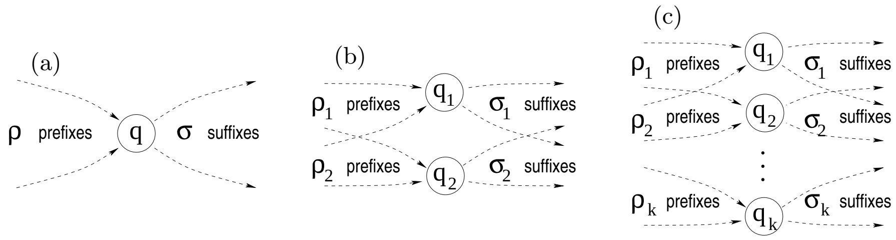

Fig. 2. Splitting of one state and its prefixes and suffixes: (a) original state, (b) splitting into two states, (c) splitting into k states.

σ_2 suffixes, such that $\varrho_1 + \varrho_2 = \varrho$ and $\sigma_1 + \sigma_2 = \sigma$, then the number of paths traversing either q_1 or q_2 is reduced to (Figure 2b):

$$\begin{aligned} \dot{p}_1 + \dot{p}_2 &= \varrho_1 \cdot \sigma_1 + \varrho_2 \cdot \sigma_2 \\ &= (\varrho - \varrho_2) \cdot \sigma_1 + (\varrho - \varrho_1) \cdot \sigma_2 \\ &= \varrho \cdot (\sigma_1 + \sigma_2) - (\varrho_2 \cdot \sigma_1 + \varrho_1 \cdot \sigma_2) \\ &= \varrho \cdot \sigma - (\varrho_2 \cdot \sigma_1 + \varrho_1 \cdot \sigma_2) \end{aligned} \tag{10}$$

$$\varrho_2 \cdot \sigma_1 + \varrho_1 \cdot \sigma_2 > 0 \quad \Rightarrow \quad \dot{p}_1 + \dot{p}_2 \; < \; \dot{p} \tag{11}$$

Splitting a state q into k states q_1 to q_k has the same effect (Figure 2c):

$$\sum_{i=1}^{k} \dot{p}_i \;=\; \varrho \cdot \sigma - \sum_{i=1}^{k} \left(\sum_{j=1, j \neq i}^{k} \varrho_j \right) \cdot \sigma_i \tag{12}$$

$$\text{with} \quad \varrho = \sum_{i=1}^{k} \varrho_i \quad \text{and} \quad \varrho = \sum_{i=1}^{k} \varrho_i$$

3. **Changing the source or destination of arcs:** The number of suffixes that follow an arc is depending on the length of the suffixes (Figure 1). If an arc leading from q_i to q_{i+1} is redirected to a following destination q_{i+m} $(m \geq 2)$ then the number of suffixes following that arc decreases and hence the total number of suffixes decreases too (Figure 3). Redirecting the arc to a preceding destination state q_{i-b} $(b \geq 0)$ would make the automaton cyclic and is therefore not in the scope of this investigation. Changing the source state of an arc will lead to similar results as changing its destination.

For an acyclic automaton with a given number of arcs, a, and a fixed length, l, the maximal number of paths, P_{max}, is reached with the festoon structure as in Figure 1 and with an as uniform as possible distribution of $\frac{a}{l}$ arcs per state on average (except for the final state). This is because any other structure can be obtained from this one by combining the three modifications above, which all reduce P_{max}.

Since this structure maximizes p for any fixed l, it does it also for variable l.

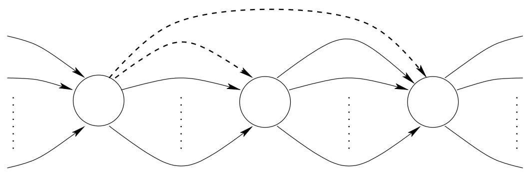

Fig. 3. Changing the destination of an arc.

4 Worst-Case Complexity for Variable Length

Let us consider a festoon, with fixed a, and variable s and l. The number n of arcs per state given by the function $out(q)$ will depend on l. However, different l will lead to different p.

The complexity of traversing and printing all paths of A, depends on the number of arcs to be "handled" (i.e., traversed or printed). This number is given by the following function:

$$f(n) \;=\; k\; p\; l \qquad\qquad \text{with} \quad k \in\;]1,2] \qquad p = n^l \qquad l = \frac{a}{n} \tag{13}$$

The coefficient $k \in\;]1, 2]$ expresses that each of the l arcs on each of the p paths is handled either once (only printed because already traversed on another path) or twice (traversed and printed). Although k is depending on n and l, we will consider it as a constant. It has no effect on the complexity of the current task.

To find the number $\widehat{n}$ of arcs per state that leads to the worst-case complexity, we compute the real $\widehat{x}$ which maximizes $f(x) = k\; l\; x^l$. We construct the first derivative of $f(x)$:

$$\begin{aligned} f(x) &= k\; x^l\; l = k\; x^{ax^{-1}}\; ax^{-1} = k\; a\; x^{ax^{-1}-1} = k\; a\; e^{(ax^{-1}-1)\ln x} \\ f(x) &= k\; a\; e^{\psi} \qquad\qquad (14) \\ &\qquad \text{with} \quad \psi = (ax^{-1}-1)\ln x \\ f'(x) &= k\; a\; \psi'\; e^{\psi} \\ f'(x) &= k\; a\; \phi\; e^{\psi} \qquad\qquad (15) \\ &\qquad \text{with} \quad \phi = \psi' = ax^{-2}(1-\ln x) - x^{-1} \end{aligned}$$

To find all extrema, we equate the first derivative to 0. Since for all k, a, ψ we have : $k\,a\,e^{\psi} > 0$, we get :

$$\begin{aligned} f'(x{=}\hat{x}) = 0 \qquad \Rightarrow \qquad \phi(x{=}\hat{x}) &= 0 \\ 0 &= a\hat{x}^{-2}(1-\ln\hat{x}) - \hat{x}^{-1} \\ \frac{1}{a} &= \frac{1-\ln\hat{x}}{\hat{x}} \end{aligned} \tag{16}$$

For an automaton with just one arc, $\hat{x}$ is obviously 1 :

$$\frac{1}{1} \; = \; \frac{1 - \ln 1}{1}$$

For large automata Equation 16 means:

$$\lim_{a \to \infty} \frac{1}{a} = 0 \qquad \Rightarrow \qquad 0 \; = \; \frac{1 - \ln \hat{x}}{\hat{x}}$$
$$\hat{x} \; = \; e \tag{17}$$

We further analyze $f'(x)$ to see for which values $x = \acute{x}$ the function $f(x)$ is growing:

$$f'(x = \acute{x}) > 0 \qquad \Rightarrow \qquad \phi(x = \acute{x}) \; > \; 0 \tag{18}$$
$$0 \; < \; a\acute{x}^{-2}(1 - \ln \acute{x}) - \acute{x}^{-1}$$
$$\frac{1}{a} \; < \; \frac{1 - \ln \acute{x}}{\acute{x}}$$
$$\lim_{a \to \infty} \frac{1}{a} = 0 \qquad \Rightarrow \qquad 0 \; < \; \frac{1 - \ln \acute{x}}{\acute{x}}$$
$$\acute{x} \; < \; e \tag{19}$$

and for which values $x = \grave{x}$ the function $f(x)$ is falling:

$$f'(x = \grave{x}) < 0 \qquad \Rightarrow \qquad \phi(x = \grave{x}) \; < \; 0 \tag{20}$$
$$0 \; > \; a\grave{x}^{-2}(1 - \ln \grave{x}) - \grave{x}^{-1}$$
$$\frac{1}{a} \; > \; \frac{1 - \ln \grave{x}}{\grave{x}}$$
$$\lim_{a \to \infty} \frac{1}{a} = 0 \qquad \Rightarrow \qquad 0 \; > \; \frac{1 - \ln \grave{x}}{\grave{x}}$$
$$\grave{x} \; > \; e \tag{21}$$

Equations 17, 19, and 21 show that $f(x)$ has its only maximum at $x = e$, is monotonically ascending for all $x < e$, and monotonically descending for all $x > e$. The maximal number of arcs to be handled is:

$$f(x = \hat{x} = e) \; = \; k \; a \; e^{ae^{-1} - 1} \; = \; \frac{k}{e} \; a \; \sqrt[e]{e}^{\; a} \tag{22}$$

Hence, the worst-case complexity of the above task, with fixed a and variable s and l is:

$$\mathcal{O}(\; f(x) \;) \; = \; \mathcal{O}(\; \sqrt[e]{e}^{\; a} \;) = \mathcal{O}(\; 1.4447^{\; a} \;) \tag{23}$$

5 Worst Case for a Given Number of States

In the previous two sections we made no assumption on s and l. In the present section s will be fixed, and l will be variable and ignored in the remainder of our analysis. This corresponds with our initial assumptions (Section 2.2).

Let A now be an automaton with fixed a and s. In this case the results of Section 3 and Section 4 may seem contradictory: s seems to impose $l = s - 1$, and s and a together seem to impose $x = \frac{a}{s-1}$, the number of arcs per state. This leads us to the question whether for $\frac{a}{l} \neq e$, the worst-case complexity is reached with $x = e$ or with uniform distribution $x = \frac{a}{s-1}$.

In fact there is no contradiction, but we have to distinguish two different cases:

1. If the number of states, s, is below the limit $s < \frac{a}{e} + 1$ then the worst case is reached with a structure as in Figure 1 and $l = s - 1$:

$$f(x) = k\,p\,l = k\,x^l\,l = k\left(\frac{a}{l}\right)^l l$$
$$= k\left(\frac{a}{s-1}\right)^{s-1}(s-1) \tag{24}$$

   This agrees with both previous results: the arcs are (approximately) uniformly distributed with $x = \frac{a}{l} > e$. To further increase the complexity, x would have to decrease towards e, which is not possible because it would require more than a states. The complexity of this case is:

$$\mathcal{O}\left(\ f(x) \middle| s - 1 < \frac{a}{e}\ \right) = \mathcal{O}\left(\ \left(\frac{a}{s-1}\right)^{s-1}\ \right) \tag{25}$$

2. If the number of states, s, exceeds the limit $s > \frac{a}{e} + 1$ then the worst case is reached with a length $l = \frac{a}{e}$, using $l + 1$ states to form a structure as in Figure 1, and the remaining states on state-splitting as in Figure 2 (Section 3, Point 2). This splitting will decrease the complexity, so that Equation 23 constitutes an (unreached) upper bound in this case:

$$\mathcal{O}\left(\ f(x) \middle| s - 1 > \frac{a}{e}\ \right) < \mathcal{O}\left(\ \sqrt[e]{e}^{\ a}\ \right) = \mathcal{O}\left(\ 1.4447^{\ a}\ \right) \tag{26}$$

   This agrees with both previous conclusions: the arcs are (approximately) uniformly distributed with $x = \frac{a}{l}$, and x equals to e, the value that maximizes the complexity.

If the number of states s equals the limit $s = \frac{a}{e} + 1$ then both previous cases hold and the equations of their complexities (25, 26) provide the same value.

6 Complexity Calculations for Some Cases

Table 1 shows results from a calculation of the function $f_a(x) = k\,p\,l$ (Equation 13), describing the task of traversing and printing a normalized acyclic automaton, given the worst-case structure, fixed a, and variable s and l. The coefficient k is set to 2. Each row gives (for fixed a) the average number of arcs per state, $\widehat{x}$, the length, $\widehat{l}$, and the number of paths, $\widehat{p}$, where $f_a(x)$ reaches its maximum:

$$k = 2 \quad (27)$$

$$\widehat{x} = \arg\max_x f_a(x) \quad (28)$$

$$\widehat{l} = \frac{a}{\widehat{x}} \quad (29)$$

$$\widehat{p} = \widehat{x}^{\widehat{l}} \quad (30)$$

$$f_a(\widehat{x}) = k\,\widehat{p}\,\widehat{l} \qquad = \max_x f_a(x) \quad (31)$$

according to Equation 13. Note that $\widehat{x} \in \mathbb{R}$ because it is an average over all states, and that in fact all $\widehat{l}, \widehat{p} \in \mathbb{N}$ rather than $\widehat{l}, \widehat{p} \in \mathbb{R}$. Thus the table gives an approximation in $\mathbb{R}$ of values that are actually in $\mathbb{N}$.

For example, in an automaton with 16 arcs ($a = 16$), the maximum is reached in fact with $l = 7$ $(\widehat{l} = 6.82)$, $x' = \frac{a}{l} = \frac{16}{7} = 2.285714$ $(\widehat{x} = 2.3474, x_i \in \{2, 3\})$, $p = 2^5 \cdot 3^2 = 288$ $(\widehat{p} = 335.7)$, and $f_a(x') = 4032$ $(f_a(\widehat{x}) = 4576)$. With growing a, $\widehat{x}$ approaches $e = 2.718282\ldots$

a	$\widehat{x}$	$\widehat{l}$	$\widehat{p}$	$f_a(\widehat{x})$
1	1.0000	1.00	1.000	2.000
2	1.3702	1.46	1.584	4.623
4	1.7535	2.28	3.601	16.43
8	2.0926	3.82	16.83	128.7
16	2.3474	6.82	335.7	4 576
32	2.5130	12.73	$1.247 \cdot 10^5$	$3.177 \cdot 10^6$
64	2.6096	24.52	$1.646 \cdot 10^{10}$	$8.073 \cdot 10^{11}$
128	2.6623	48.08	$2.791 \cdot 10^{20}$	$2.684 \cdot 10^{22}$
256	2.6898	95.17	$7.913 \cdot 10^{40}$	$1.506 \cdot 10^{43}$
512	2.7040	189.35	$6.311 \cdot 10^{81}$	$2.390 \cdot 10^{84}$
1024	2.7112	377.70	$3.998 \cdot 10^{163}$	$3.020 \cdot 10^{166}$
2048		*program numeric overflow*		

Table 1. Calculation of the worst-case complexity with fixed a and variable s and l.

7 Conclusion

Our investigation has shown (Equations 25, 26) that the complexity of traversing and printing all paths of a normalized acyclic automaton with s states and a arcs reaches its maximum with a structure as in Figure 1 and (approximately) uniform distribution of arcs over the states (except for the final state that has no outgoing arcs). For large a, and depending on s, the worst-case complexity is:

$$\mathcal{O}(f(x)) = \begin{cases} \mathcal{O}\left(\left(\frac{a}{s-1}\right)^{s-1}\right) & \text{for} \quad a > e(s-1) \\ \mathcal{O}(\ \sqrt[e]{e}^{\ a}\ ) = \mathcal{O}\left(e^{\frac{a}{e}}\right) < \mathcal{O}\left(e^{s-1}\right) & \text{for} \quad a \leq e(s-1) \end{cases}$$

Acknowledgments. We would like to thank Jean-Marc Champarnaud for his advice.

References

Berstel, J. 1989. Finite Automata and Rational Languages. An Introduction, in Formal Properties of Finite Automata and Applications. In J.-E. Pin, editor, *Lecture Notes in Computer Science*, number 386. Verlag, 1989 edition, pages 2–14.

Caron, P. and D. Ziadi. 2000. Characterization of Glushkov automata. *Theoret. Comput. Sci.*, 233(1–2):75–90.

Eilenberg, S. 1974. *Automata, Languages, and Machines*, volume A. Academic Press, San Diego, CA, USA.

Giammarresi, D., J.-L. Ponty, and D. Wood. 2001. Thompson digraphs: A characterization, in WIA'99. *Lecture Notes in Computer Science*, 2214:91–100.

Hopcroft, J. E, R. Motwani, and J. D Ullman. 2001. *Introduction to Automata Theory, Languages and Computation*. Low Price Edition. Addison Wesley Longman, Inc, Reading, Mass., USA, 2 edition.

Nicaud, C. 2000. Étude du comportement en moyenne des automates finis et des languages rationnels. Thesis, University of Paris 7.

Perrin, D. 1990. Finite automata. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science: Volume B: Formal Models and Semantics*. Elsevier, Amsterdam, pages 1–57.

Yu, S., Q. Zhuang, and K. Salomaa. 1994. The state complexities of some basic operations on regular languages. *Theoret. Comput. Sci.*, 125(2):315–328.