Fundamenta Informaticae 144 (2016) 161–182
DOI 10.3233/FI-2016-1326
IOS Press

State Complexity of Basic Operations on Non-Returning Regular Languages

Hae-Sung Eom, Yo-Sub Han*

Department of Computer Science, Yonsei University
50, Yonsei-Ro, Seodaemun-Gu, Seoul 120-749, Republic of Korea
{haesung, emmous}@cs.yonsei.ac.kr

Galina Jirásková

Mathematical Institute, Slovak Academy of Sciences
Grešákova 6, 040 01 Košice, Slovakia
jiraskov@saske.sk

Abstract. We consider the state complexity of basic operations on non-returning regular languages. For a non-returning minimal DFA, the start state does not have any in-transitions. We establish the precise state complexity of four Boolean operations (union, intersection, difference, symmetric difference), catenation, reverse, and Kleene-star for non-returning regular languages. Our results are usually smaller than the state complexities for general regular languages and larger than the state complexities for suffix-free regular languages. In the case of catenation and reversal, we define witness languages over a ternary alphabet. Then we provide lower bounds for a binary alphabet. For every operation, we also study the unary case.

Keywords: Finite automata, non-returning regular languages, basic operations, state complexity.

1. Introduction

Given a regular language L, researchers often use the number of states in the minimal deterministic finite-state automaton (DFA) for L to represent the complexity of L. Based on this notion, the state complexity of an operation for regular languages is defined as the number of states that are necessary

*Address for correspondence: Department of Computer Science, Yonsei University, 50, Yonsei-Ro, Seodaemun-Gu, Seoul 120-749, Republic of Korea.

Received February 2015; revised January 2016.

and sufficient in the worst-case for the minimal DFA to accept the language resulting from the operation, considered as a function of the state complexities of operands.

Maslov [1] provided, without giving proofs, the state complexity of union, catenation, and star, and later Yu et al. [2] investigated the state complexity further. The state complexity of an operation is calculated based on the structural properties of input regular languages and a given operation. Recently, due to large amount of memory, fast CPUs and massive data size, many applications using regular languages require finite-state automata (FAs) of very large size. This makes the estimated upper bound of the state complexity useful in practice since it helps to manage resources efficiently. Moreover, it is a challenging quest to verify whether or not an estimated upper bound can be reached.

Yu [3] gave a comprehensive survey of the state complexity of regular languages. Salomaa et al. [4] studied classes of languages, for which the reverse operation reaches the exponential upper bound. As special cases of the state complexity, researchers examined the state complexity of finite languages [5, 6], the state complexity of unary language operations [7] and the nondeterministic descriptional complexity of regular languages [8]. There are several other results with respect to the state complexity of different operations [9, 10, 11, 12, 13, 14].

For regular language codes, which preserve certain structural properties in the corresponding minimal DFAs, Han et al. [15] studied the state complexity of prefix-free regular languages. Similarly, based on suffix-freeness, Han and Salomaa [16] looked at the state complexity of suffix-free regular languages. Note that a prefix-free minimal DFA has a single final state and all out-transitions of the final state go to the sink state [17]. Moreover, this property is the necessary and sufficient condition for a minimal DFA A to be prefix-free; namely, $L(A)$ is prefix-free. For a suffix-free minimal DFA, the start state does not have any in-transitions [16]. A DFA with this property is called non-returning. However, this non-returning property is only a necessary condition for a minimal DFA to be suffix-free, but it is not sufficient. This implies that all subfamilies of suffix-free regular languages— such as bifix-, infix-, or outfix-free regular languages—have the non-returning property. Note that finite languages also preserve this property. In addition, some DFAs in model checking systems are non-returning as well. These DFAs start with a specific pre-condition in model checking; once they read the pre-condition at the start state, they never visit the start state again in the remaining process. These observations intrigue us to investigate DFAs with non-returning property and the state complexity of basic operations on languages accepted by non-returning DFAs. Notice that the state complexity of operations on fundamental subfamilies of the regular languages can provide valuable insights on connections between restrictions placed on language definitions and descriptional complexity.

We start with an observation that a non-returning language and its complement have the same complexity. We get the tight bound $(m-1)(n-1)+1$ for four Boolean operations (union, intersection, difference, symmetric difference). To prove tightness, we use a binary alphabet. We also show that this bound is tight in the unary case if $m-1$ and $n-1$ are relatively prime numbers, except for symmetric difference, where the tight bound is $(m-1)(n-1)$ if $m-1$ and $n-1$ are relatively prime.

In the case of catenation, we get the tight bound $(m-1)2^{n-1}+1$, and our witnesses are defined over a ternary alphabet. In the binary case, we are still able to prove an exponential lower bound $(m-2)2^{n-1}+2^{n-2}+2$. For catenation on unary non-returning languages, we get an upper bound mn, and a lower bound $(m-1)(n-1)+2$ if $m-1$ and $n-1$ are relatively prime.

We next study the reversal operation on non-returning languages. Here we get the tight bound 2^n which is the same as in the general case of regular languages. However, to define worst-case examples, we use a ternary alphabet, while in the general case, there exist binary witness languages. We conjecture that the bound 2^n cannot be met by any binary non-returning language. On the other hand, we still have an exponential lower bound 2^{n-2} in the binary case. The reversal operation on unary languages is trivial since the reversal of any unary language is the same language.

We conclude our paper with the Kleene star operation. We get the tight bound 2^{n-1} for any alphabet with at least two symbols. Then we show that in the unary case, the tight bound is $(n-2)^2+2$. All our results are usually less than the state complexities for general regular languages and greater than the state complexities for suffix-free regular languages.

In Section 2, we define some basic notions and prove preliminary results. Then we formally define non-returning regular languages. We prove the tight bounds on the state complexity of complementation, Boolean operations, catenation, reversal, and Kleene star in Sections 3, 4, 5, 6, and 7, respectively. We summarize the state complexity results and compare them with the regular language case and the suffix-free case in Section 8.

2. Preliminaries

Let Σ denote a finite alphabet of characters, and Σ^* denote the set of all strings over Σ. The size $|\Sigma|$ of Σ is the number of characters in Σ. A language over Σ is any subset of Σ^*. The symbol $\emptyset$ denotes the empty language and the symbol λ denotes the null string. Let $|w|_a$ be the number of a appearances in a string w. For strings x, y and z, we say that y is a *suffix* of z if $z = xy$. We define a language L to be suffix-free if for any two distinct strings x and y in L, x is not a suffix of y. For a string x, let x^R be the reverse of x and for a language L, we denote $L^R = \{x^R \mid x \in L\}$.

A DFA A is specified by a tuple $(Q, \Sigma, \delta, s, F)$, where Q is a finite set of states, Σ is an input alphabet, $\delta : Q \times \Sigma \to Q$ is a transition function, $s \in Q$ is the start state and $F \subseteq Q$ is a set of final states. The *state complexity* of a regular language L, $\mathrm{sc}(L)$, is defined to be the number of states of the minimal DFA recognizing L.

Given a DFA A, we assume that A is complete; therefore, A may have a sink state. For a transition $\delta(p, a) = q$ in A, we say that p has an *out-transition* and q has an *in-transition*. We say that A is *non-returning* if the start state of A does not have any in-transitions. We define a regular language to be a *non-returning regular language* if its minimal DFA is non-returning.

A nondeterministic finite automaton (NFA) is a tuple $M = (Q, \Sigma, \delta, Q_0, F)$ where Q, Σ, F are as in a DFA, Q_0 is the set of start states, and $\delta\colon Q \times \Sigma \to 2^Q$ is the transition function. Every NFA M can be converted to an equivalent DFA $M' = (2^Q, \Sigma, \delta', Q_0, F')$ by the subset construction. We call the DFA M' the *subset automaton* of the NFA M.

For complete background knowledge in automata theory, we refer to the textbooks [18, 19, 20]. To conclude this section let us state some preliminary results that we will use later throughout the paper.

Proposition 2.1. Let N be an NFA such that for every state q, there exists a string w_q accepted by the NFA N from state q and rejected from any other state. Then all states of the subset automaton of N are pairwise distinguishable.

Proof:
Let S and T be two distinct subsets of the subset automaton. Then, without loss of generality, there is a state q of N such that $q \in S$ and $q \notin T$. Then the string w_q is accepted by the subset automaton from S and rejected from T. $\square$

The following well-known observation allows us to avoid the proof of distinguishability in the case of reverse. It can be easily proved using Proposition 2.1, and for the sake of completeness, we present the proof here.

Proposition 2.2. ([21])
All states of the subset automaton of the reverse of a minimal DFA are pairwise distinguishable.

Proof:
Let A be a minimal DFA. Since every state of A is reachable, for every state q of the NFA A^R, there exists a string w_q that is accepted by A^R from q. Since A is deterministic, the string w_q cannot be accepted by A^R from any other state. Hence the NFA A^R satisfies the condition of Proposition 2.1, and therefore all states of the subset automaton of A^R are pairwise distinguishable. □

If N is a non-returning NFA with the state set Q and the initial state s, then the only reachable subset of the subset automaton of N containing the state s is $\{s\}$. If, moreover, the empty set is unreachable in the subset automaton, then two distinct subsets of the subset automaton must differ in a state from $Q \setminus \{s\}$. Hence a sufficient condition for distinguishability in such a case is as follows.

Proposition 2.3. Let $N = (Q, \Sigma, \delta, s, F)$ be a non-returning NFA such that the empty set is unreachable in the corresponding subset automaton. Assume that for every state q in $Q \setminus \{s\}$, there exists a string w_q accepted by N only from q. Then all states of the subset automaton of N are pairwise distinguishable.

3. Complement

Let us start with the complementation operation on non-returning languages. If L is a language over an alphabet Σ, then the complement of L is the language $L^c = \Sigma^* \setminus L$. To get a DFA for the complement of a given regular language, we only need to interchange the final and non-final states in a DFA for the given language. Formally, if a regular language L is accepted by a DFA $A = (Q, \Sigma, \delta, s, F)$, then the language L^c is accepted by the DFA $A^c = (Q, \Sigma, \delta, s, Q \setminus F)$. Moreover, if A is minimal, then A^c is minimal as well. It follows that the state complexity of a regular language and its complement is the same. Next, notice that if a DFA A is non-returning, then the DFA A^c is also non-returning. Hence we have the following result.

Theorem 3.1. Let L be a non-returning regular language. Then $\mathrm{sc}(L) = \mathrm{sc}(L^c)$.

4. Boolean Operations

Now we consider the following four Boolean operations: intersection, union, difference, and symmetric difference. In the general case of all regular languages, the state complexity of all four operations is given by the function mn, and the worst case examples are defined over a binary alphabet [2, 22]. In the case of non-returning languages, we obtain the precise state complexity for these operations, which again turn out to be the same, except for symmetric difference in the case of $m = n = 2$. Let us start with intersection.

Lemma 4.1. Let K and L be non-returning regular languages over an alphabet Σ with $\mathrm{sc}(K) = m$ and $\mathrm{sc}(L) = n$, where $m, n \geq 2$. Then $\mathrm{sc}(K \cap L) \leq (m-1)(n-1)+1$, and the bound is tight if $|\Sigma| \geq 2$.

Proof:
Let K and L be accepted by a non-returning m-state and n-state DFA, respectively. Let the state sets of the two DFAs be Q_A and Q_B, and let the start states be s_A and s_B, respectively. Construct the product automaton for $K \cap L$ with the state set $Q_A \times Q_B$. Since both DFAs are non-returning, in the product automaton, all the states (s_A, q) and (p, s_B), except for the initial state (s_A, s_B), are unreachable. This gives the upper bound.

To prove tightness, let

$$K = \{(a+b)\,w \mid w \in \{a,b\}^* \text{ and } |w|_a \geq m-2\},$$
$$L = \{(a+b)\,w \mid w \in \{a,b\}^* \text{ and } |w|_b \geq n-2\}.$$

The languages K and L are accepted by the non-returning DFAs shown in Figure 1.

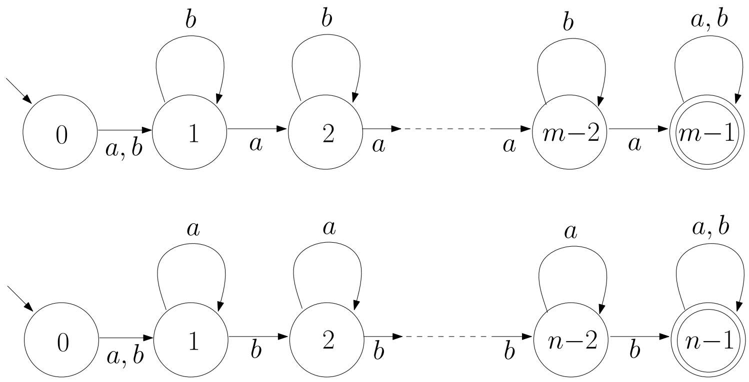

Figure 1. The binary non-returning witnesses for intersection meeting the bound $(m-1)(n-1)+1$.

In the product automaton for the language $K \cap L$, the unique final state is $(m-1, n-1)$. The product automaton in the case of $m = 4$ and $n = 5$ is shown in Figure 2. The state $(1,1)$ is reached from the initial state $(0,0)$ by a. Every state (i,j) with $1 \leq i \leq m-1$ and $1 \leq j \leq n-1$ is reached from $(1,1)$ by $a^{i-1}b^{j-1}$. This proves the reachability of $(m-1)(n-1)+1$ states.

Now let (i,j) and (k,ℓ) be two distinct states of the product automaton. If $i < k$, then the string $a^{m-1-k}b^n$ is accepted from (k,ℓ) and rejected from (i,j). If $j < \ell$, then the string $b^{n-1-\ell}a^m$ is accepted from (k,ℓ) and rejected from (i,j). This proves distinguishability. □

Now we are going to prove the tight bounds for union and difference. We use the lemma above, the equalities $K \cup L = (K^c \cap L^c)^c$ and $K \setminus L = K \cap L^c$, and the fact that the state complexity of a regular language is the same as the state complexity of its complement.

Lemma 4.2. Let K and L be non-returning regular languages over an alphabet Σ with $\mathrm{sc}(K) = m$ and $\mathrm{sc}(L) = n$, where $m, n \geq 2$. Then $\mathrm{sc}(K \cup L), \mathrm{sc}(K \setminus L) \leq (m-1)(n-1)+1$, and the bound is tight if $|\Sigma| \geq 2$.

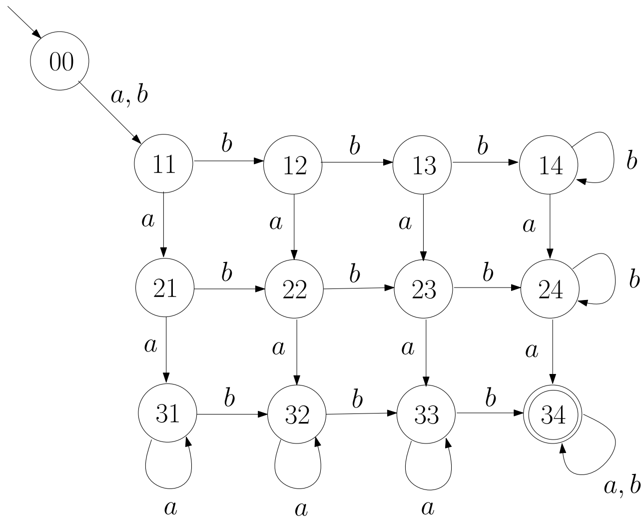

Figure 2. The product automaton for intersection of languages from Figure 1; $m = 4, n = 5$.

Proof:
We prove the upper bound in the same way as for intersection. To prove the tightness for union, consider the languages K^c and L^c, where K and L are the binary witness languages for intersection described in the proof of the previous lemma. The languages K^c and L^c are non-returning with state complexities m and n, respectively. Since $K^c \cup L^c = (K \cap L)^c$, we have $\mathrm{sc}(K^c \cup L^c) = (m-1)(n-1)+1$. For difference, we take the languages K and L^c. Since $K \setminus L^c = K \cap L$, we have $\mathrm{sc}(K \setminus L^c) = (m-1)(n-1)+1$. $\square$

Finally, we consider symmetric difference on non-returning regular languages. Here we must be careful with the case of $m = n = 2$.

Lemma 4.3. Let K and L be non-returning regular languages over an alphabet Σ with $\mathrm{sc}(K) = m$ and $\mathrm{sc}(L) = n$, where $m, n \geq 2$. If $m = n = 2$, then $\mathrm{sc}(K \oplus L) = 1$. Otherwise, $\mathrm{sc}(K \oplus L) \leq (m-1)(n-1)+1$, and the bound is tight if $|\Sigma| \geq 2$.

Proof:
If a regular language over an alphabet Σ is non-returning, and has state complexity 2, then it is equal to λ or $\Sigma\Sigma^*$. The symmetric difference of two such languages is the empty language, so its state complexity is 1.

Otherwise, we get the upper bound in the same way as for intersection. To prove tightness, we take the same languages as for intersection. In the product automaton, the final states are $(i, n-1)$ with $1 \leq i \leq m-2$ and $(m-1, j)$ with $1 \leq j \leq n-2$. The proof of reachability is the same as in the case of intersection. If $i < k$ then the string $a^{m-1-k}b^n$ is rejected from (k, ℓ) and accepted from (i, j). If $j < \ell$, then the string $b^{n-1-\ell}a^m$ is rejected from (k, ℓ) and accepted from (i, j). This completes the proof. $\square$

Now we consider the unary case. We show that the upper bound $(m-1)(n-1)+1$ for intersection, union, and difference can be met whenever $\gcd(m-1, n-1) = 1$.

Lemma 4.4. Let $m, n \geq 2$ and $\circ \in \{\cap, \cup, \setminus\}$. Let K and L be unary non-returning regular languages with $\mathrm{sc}(K) = m$ and $\mathrm{sc}(L) = n$. Then $\mathrm{sc}(K \circ L) \leq (m-1)(n-1)+1$, and the bound is tight if $\gcd(m-1, n-1) = 1$.

Proof:
The upper bound follows from the fact that all the states in the first row and the first column of the product automaton are unreachable, except for the initial state.

To prove tightness, we first consider intersection. Let $K = a^{m-1}(a^{m-1})^*$ and $L = a^{n-1}(a^{n-1})^*$. Then K and L are non-returning with $\mathrm{sc}(K) = m$ and $\mathrm{sc}(L) = n$. Since $\gcd(m-1, n-1) = 1$, we have $K \cap L = a^{(m-1)(n-1)}(a^{(m-1)(n-1)})^*$. Therefore $\mathrm{sc}(K \cap L) = (m-1)(n-1)+1$.

The same result for union and difference follows from the results for complementation and intersection, and and the equalities $K \cup L = (K^c \cap L^c)^c$ and $K \setminus L = K \cap L^c$. $\square$

Finally, consider symmetric difference on unary non-returning languages. The next lemma shows that the upper bound on the state complexity of symmetric difference on unary non-returning languages is $(m-1)(n-1)$ whenever $m, n \geq 3$. We also prove that this bound is tight if $m-1$ and $n-1$ are relatively prime numbers.

Lemma 4.5. Let $m, n \geq 3$. Let K and L be unary non-returning regular languages with $\mathrm{sc}(K) = m$ and $\mathrm{sc}(L) = n$. Then $\mathrm{sc}(K \oplus L) \leq (m-1)(n-1)$, and the bound is tight if $\gcd(m-1, n-1) = 1$.

Proof:
Let the languages K and L be accepted by unary minimal non-returning deterministic finite automata $A = (\{0, 1, \ldots, m-1\}, \{a\}, \delta_A, 0, F_A)$ and $B = (\{0, 1, \ldots, n-1\}, \{a\}, \delta_B, 0, F_B)$, respectively. Since A and B are non-returning, in the product automaton for symmetric difference, all states $(0, q)$ and $(q, 0)$ are unreachable, except for the state $(0, 0)$. This gives the upper bound $(m-1)(n-1)+1$.

If $\delta_A(m-1, a) \geq 2$ or $\delta_B(n-1, a) \geq 2$, then at least one more state $(1, p)$ or $(q, 1)$ is unreachable in the product automaton since $m, n \geq 3$. Hence the upper bound is $(m-1)(n-1)$ in this case.

Now assume that $\delta_A(m-1, a) = 1$ and $\delta_B(n-1, a) = 1$. Since A and B are minimal, the states 0 and $m-1$ of A, as well as the states 0 and $n-1$ of B, do not have the same finality [12, Lemma 1]. It follows that the states $(0, 0)$ and $(m-1, n-1)$ of the product automaton for $K \oplus L$ have the same finality, and therefore can be merged. Thus, also in this case, we get the upper bound $(m-1)(n-1)$.

To prove tightness, let $\gcd(m-1, n-1) = 1$. Then by the Chinese Remainder Theorem, for every integers i, j, there is a solution $x(i, j)$ to the following simultaneous congruences:

$$x(i, j) \equiv i \pmod{m-1}, \tag{1}$$
$$x(i, j) \equiv j \pmod{n-1}. \tag{2}$$

Let $K = a^{m-1}(a^{m-1})^*$ and $L = a^{n-1}(a^{n-1})^*$. Then K and L are accepted by the unary non-returning minimal automata A and B shown in Figure 3.

Construct the product automaton for the language $K \oplus L$. The initial state of the product automaton is (s_A, s_B). The final states of the product automaton are $(m-2, j)$ and $(i, n-2)$, except for $(m-2, n-2)$. Let us show that all the states (i, j), where $0 \leq i \leq m-2$ and $0 \leq j \leq n-2$, are reachable and pairwise distinguishable in the product automaton for the language $K \oplus L$.

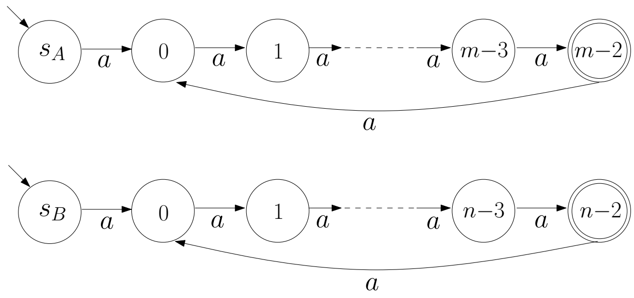

Figure 3. The unary non-returning witnesses for symmetric difference meeting the bound $(m-1)(n-1)$; $m, n \geq 3$ and $\gcd(m-1, n-1) = 1$.

Since we have

$$(s_A, s_B) \xrightarrow{a} (0,0) \xrightarrow{a^{x(i,j)}} (i,j),$$

each state (i,j) is reachable; recall that $x(i,j)$ is the solution of (1) and (2).

Now, we prove distinguishability.

First, consider two final states in the last row of the subset automaton, that is, two states $(m-2, j)$ and $(m-2, \ell)$, where $0 \leq j < \ell \leq n-3$. Then we have

$$(m-2, \ell) \xrightarrow{a^{x(0,n-2-\ell)}} (m-2, n-2),$$
$$(m-2, j) \xrightarrow{a^{x(0,n-2-\ell)}} (m-2, n-2-(\ell-j)).$$

Hence the string $a^{x(0,n-2-\ell)}$ is rejected from $(m-2,\ell)$, but accepted from $(m-2,j)$. The case of two final states in the last column of the product automaton is symmetric: two states $(i, n-2)$ and $(k, n-2)$, where $0 \leq i < k \leq m-3$, are distinguished by $a^{x(m-2-k,0)}$.

Now, consider a final state $(m-2, j)$ in the last row and a final state $(i, n-2)$ in the last column of the product automaton. If $j \leq n-4$, then we have

$$(m-2, j) \xrightarrow{a^{x(0,1)}} (m-2, j+1),$$
$$(i, n-2) \xrightarrow{a^{x(0,1)}} (i, 0).$$

Hence the string $a^{x(0,1)}$ is accepted from $(m-2, j)$, but rejected from $(i, n-2)$. If $j = n-3$, then the string $a^{x(0,2)}$ distinguishes the two states since we have

$$(m-2, n-3) \xrightarrow{a^{x(0,2)}} (m-2, 0),$$
$$(i, n-2) \xrightarrow{a^{x(0,2)}} (i, 1).$$

We have shown that final states of the product automaton are pairwise distinguishable.

Finally, consider two distinct non-final states (i,j) and (k,ℓ). By the string $a^{x(m-2-i,n-3-j)}$, the state (i,j) goes to the accepting state $(m-2, n-3)$. On the other hand, the state (k,ℓ) goes by $a^{x(m-2-i,n-3-j)}$ either to a rejecting state, or to an accepting state different from $(m-2, n-3)$. It follows that (i,j) and (k,ℓ) are distinguishable. □

The next theorem summarizes our results on Boolean operations.

Theorem 4.6. (Boolean Operations) Let $m, n \geq 2$ and $\circ \in \{\cap, \cup, \setminus, \oplus\}$. Let $f_k^\circ(m,n)$ be the state complexity of the $\circ$ operation on non-returning regular languages over a k-letter alphabet defined as $f_k^\circ(m,n) = \max\{\mathrm{sc}(K \circ L) \mid K, L \subseteq \Sigma^*, |\Sigma| = k, \mathrm{sc}(K) = m, \mathrm{sc}(L) = n,$ and K, L non-returning$\}$. Then

(*i*) if $k \geq 2$ and $\circ \in \{\cap, \cup, \setminus\}$, then $f_k^\circ(m,n) = (m-1)(n-1)+1$.

(*ii*) if $k \geq 2$, then $f_k^\oplus(m,n) = \begin{cases} 1, & \text{if } m = n = 2; \\ (m-1)(n-1)+1, & \text{otherwise.} \end{cases}$

(*iii*) if $\circ \in \{\cap, \cup, \setminus\}$, then
$f_1^\circ(m,n) \leq (m-1)(n-1)+1$,
$f_1^\circ(m,n) = (m-1)(n-1)+1$ if $\gcd(m-1, n-1) = 1$.

(*iv*) $f_1^\oplus(2,2) = 1$,
$f_1^\oplus(2,n) = n$, where $n \geq 3$,
$f_1^\oplus(m,2) = m$, where $m \geq 3$,
$f_1^\oplus(m,n) \leq (m-1)(n-1)$, where $m, n \geq 3$,
$f_1^\oplus(m,n) = (m-1)(n-1)$, where $m, n \geq 3$ and $\gcd(m-1, n-1) = 1$.

Proof:
(*i*) The tight bounds for intersection, union, and difference for alphabets of at least two symbols are given by Lemmas 4.1 and 4.2. (*ii*) The tight bound for symmetric difference for alphabets of at least two symbols is given by Lemma 4.3. (*iii*) The tightness follows from Lemma 4.4. (*iv*) If $m = n = 2$, then the languages are either $\{\lambda\}$ or aa^*, and their symmetric difference is the empty language. If $m = 2$ and $n \geq 3$, then the upper bound is $1 \cdot (n-1) + 1$, and it is met by the symmetric difference of the languages aa^* and $a^{n-1}(a^{n-2})^*$. The case of $m \geq 3$ and $n = 2$ is symmetric. The upper bound in the case of $m, n \geq 3$, as well as its tightness, whenever $\gcd(m-1, n-1) = 1$, is given by Lemma 4.5. □

5. Catenation

The state complexity of catenation on regular languages is given by the function

$$f(m,n) = \begin{cases} m, & \text{if } m \geq 1 \text{ and } n = 1; \\ m2^n - 2^{n-1}, & \text{if } m \geq 1 \text{ and } n \geq 2. \end{cases}$$

and the worst case examples can be defined over a binary alphabet [1, 2, 23, 24]. In this section we give the tight bound for catenation on non-returning languages.

We start with the case when the state complexity of the second language is two.

Lemma 5.1. Let $m \geq 2$. Let K and L be non-returning regular languages over an alphabet Σ with $\mathrm{sc}(K) = m$ and $\mathrm{sc}(L) = 2$. Then $\mathrm{sc}(KL) \leq m+1$, and the bound is tight if $|\Sigma| \geq 1$.

Proof:
If L is a non-returning language over an alphabet Σ with $\mathrm{sc}(L) = 2$, then $L = \{\lambda\}$ or $L = \Sigma\Sigma^*$.

If $L = \{\lambda\}$, then $KL = K$, so $\mathrm{sc}(KL) = m$.

If $L = \Sigma\Sigma^*$, then $KL = K\Sigma\Sigma^*$. Let A be a DFA for K. To get a DFA for $K\Sigma\Sigma^*$ from the DFA A, we add a new final state f which goes to itself on every symbol in Σ. Next, we remove all out-transitions of all final states in A, and we add the transitions from every final state in A to state f on each symbol in Σ. This gives the upper bound $m + 1$.

For tightness, we consider the unary non-returning regular languages $K = \{a^i \mid i \geq m - 1\}$ and $L = aa^*$ with $\mathrm{sc}(K) = m$ and $\mathrm{sc}(L) = 2$. Then $KL = Kaa^* = \{a^i \mid i \geq m\}$, so we have $\mathrm{sc}(KL) = m + 1$. □

The following lemma provides an upper bound on the state complexity of catenation on non-returning regular languages in all the remaining cases.

Lemma 5.2. Let $m \geq 2$ and $n \geq 3$. Let K and L be non-returning languages over an alphabet Σ with $\mathrm{sc}(K) = m$ and $\mathrm{sc}(L) = n$. Then $\mathrm{sc}(KL) \leq (m - 1)2^{n-1} + 1$.

Proof:
Let $A = (Q_A, \Sigma, \delta_A, s_A, F_A)$ and $B = (Q_B, \Sigma, \delta_B, s_B, F_B)$ be minimal non-returning DFAs for K and L with m and n states, respectively.

Construct an NFA N for the language KL from the DFAs A and B by adding a transition on every symbol a in Σ from every final state of A to the state $\delta(s_B, a)$, and by omitting the state s_B. The initial state of N is s_A and the set of final states is F_B. Moreover, the NFA N is non-returning.

Apply the subset construction to the NFA N. Since the automaton A is deterministic, every reachable state of the subset automaton contains exactly one state of the DFA A and, possibly, some states of the DFA B, except for the state s_B. Moreover, the only subset containing the state s_A is $\{s_A\}$. It follows that the subset automaton has at most $(m - 1)2^{n-1} + 1$ reachable states, which proves the upper bound. □

Now we prove that the upper bound given by Lemma 5.2 is tight. First, we consider the case of $m = 2$ and $n \geq 3$, and prove the tightness of the bound $2^{n-1} + 1$. Notice that we need a growing alphabet in this case.

Lemma 5.3. Let $n \geq 3$. There exist non-returning regular languages K and L over an alphabet Σ with $|\Sigma| = n - 1$ such that $\mathrm{sc}(K) = 2$, $\mathrm{sc}(L) = n$, and $\mathrm{sc}(KL) = 2^{n-1} + 1$. The bound $2^{n-1} + 1$ cannot be met for smaller alphabets.

Proof:
Let $n \geq 3$ and $\Sigma = \{a_0, a_1, \ldots, a_{n-2}\}$. Let $K = \Sigma\Sigma^*$. Then the language K is accepted by the minimal two-state non-returning DFA $A = (\{s_A, q_0\}, \Sigma, s_A, \delta_A, \{q_0\})$, in which $\delta_A(s_A, a) = \delta_A(q_0, a) = q_0$ for each a in Σ.

Next, we consider the regular language L accepted by the minimal n-state non-returning DFA $B = (\{s_B, 0, 1, \ldots, n-2\}, \Sigma, \delta, s_B, \{n-2\})$, in which

$\delta_B(s_B, a_i) = i$,

$\delta_B(i, a_0) = (i + 1) \bmod (n - 1)$ and

$\delta_B(i, a_j) = i$,
for $i = 0, 1, \ldots, n-2$ and $j = 1, 2, \ldots, n-2$, that is, the initial state s_B goes to state i on symbol a_i, there is a cycle $(0, 1, \ldots, n-2)$ on symbol a_0, and each state i goes to itself on symbol a_j with $j \geq 1$. The DFAs A and B are shown in Figure 4.

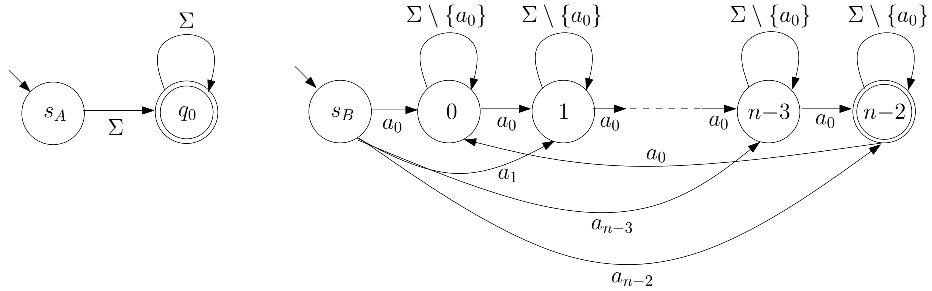

Figure 4. The non-returning witnesses for catenation meeting the bound $2^{n-1} + 1$; $m = 2$, $n \geq 3$, and $\Sigma = \{a_0, a_1, \ldots, a_{n-2}\}$.

Construct an NFA N for the language KL from DFAs A and B by adding the transitions on each symbol a_i from state q_0 to state i, and by omitting the state s_B. The initial state of N is s_A, and the final state is $n-2$. Let us show that the subset automaton of the NFA N has $2^{n-1} + 1$ reachable and pairwise distinguishable states.

The initial state of the subset automaton is $\{s_A\}$, and it goes to state $\{q_0\}$ by a_0. Next, notice that each set $\{q_0\} \cup \{0, i_2, \ldots, i_k\}$, where $1 \leq i_2 < \cdots < i_k \leq n-2$, is reached from the set $\{q_0\} \cup \{i_2 - 1, \ldots, i_k - 1\}$ by a_0, and each set $\{q_0\} \cup \{i_1, i_2, \ldots, i_k\}$, where $1 \leq i_1 < i_2 < \cdots < i_k \leq n-2$, is reached from the set $\{q_0\} \cup \{i_2, \ldots, i_k\}$ by a_{i_1}. This proves the reachability of all the sets $\{q_0\} \cup S$ with $S \subseteq \{0, 1, \ldots, n-2\}$ by induction.

Now we prove distinguishability. The initial state $\{s_A\}$ and a state $\{q_0\} \cup S$ can be distinguished by a_0^{n-1} which is accepted by N from q_0 but rejected from s_A. Two distinct states $\{q_0\} \cup S$ and $\{q_0\} \cup T$ differ in a state j with $0 \leq j \leq n-2$, and the string a_0^{n-2-j} distinguishes the two states.

Finally, let us show that the bound $2^{n-1} + 1$ cannot be met for smaller alphabets. Notice that each reachable state $q_0 \cup S$, where S is a non-empty subset of $\{0, 1, \ldots, n-2\}$, must contain at least one of the states $\delta_B(s_B, a)$ with a in Σ. If $|\Sigma| < n-2$, then the set $\{0, 1, \ldots, n-2\} \setminus \{\delta_B(s_B, a) \mid a \in \Sigma\}$ is non-empty, and no subset of this set is reachable in the subset automaton of the NFA N. Our proof is complete. □

Our next result shows that the upper bound given by Lemma 5.2 is tight for alphabets with at least three symbols in the case of $m, n \geq 3$.

Lemma 5.4. Let $m, n \geq 3$. There exist ternary non-returning regular languages K and L with $\mathrm{sc}(K) = m$ and $\mathrm{sc}(L) = n$, and such that $\mathrm{sc}(KL) = (m-1)2^{n-1} + 1$.

Proof:
Let $m, n \geq 3$. Let K and L be the ternary non-returning languages accepted by the DFAs A and B shown in Figure 5.

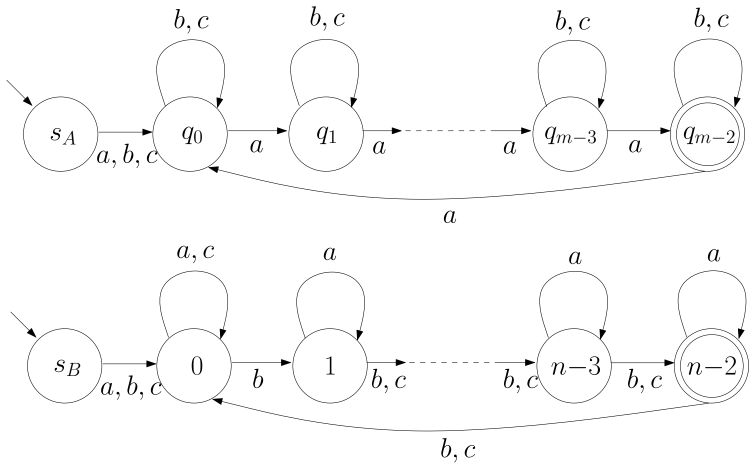

Figure 5. The ternary non-returning witnesses for catenation meeting the bound $(m-1)2^{n-1}+1$; $m,n \geq 3$.

Construct an NFA N for KL from the DFAs A and B by adding transitions on a, b, c from the state q_{m-2} to state 0, and by omitting state s_B. The initial state of N is s_A, and the unique final state is $n-2$. Let us show that the subset automaton of the NFA N has $(m-1)2^{n-1}+1$ reachable and pairwise distinguishable states.

We prove by induction that every set $\{q_i, j_1, j_2, \ldots, j_k\}$, where $0 \leq i \leq m-2$ and $0 \leq j_1 < j_2 < \cdots < j_k \leq n-2$, is reachable from the initial state $\{s_A\}$.

The basis, $k=0$, holds since $\{q_i\}$ is reached from $\{s_A\}$ by a^{i+1}. Let $1 \leq k \leq n-2$, and assume that the claim holds for $k-1$. Let $S = \{q_i, j_1, j_2, \ldots, j_k\}$. Consider the following three cases:

(*i*) $i = 0$ and $j_1 = 0$. Let $S' = \{q_{m-2}, j_2, \ldots, j_k\}$. Then S' is reachable by the induction hypothesis. Since S' goes to S by a, the set S is reachable;

(*ii*) $i = 0$ and $j_1 \geq 1$. Let $S' = \{q_0, 0, j_2 - j_1, \ldots, j_k - j_1\}$. Then S' is reachable as shown in case (*i*), and goes to S by b^{j_1};

(*iii*) $i \geq 1$. Let $S' = \{q_0, j_1, j_2, \ldots, j_k\}$. Then S' is reachable as shown in cases (*i*) and (*ii*), and goes to S by a^i.

To prove distinguishability, let $0 \leq j \leq n-2$ and $0 \leq i \leq m-3$. The string b^{n-2-j} is accepted by the NFA N only from the state j, the string $c^n b \cdot b^{n-2}$ is accepted only from q_{m-2}, and the string $a^{m-2-i} \cdot c^n b \cdot b^{n-2}$ is accepted only from q_i. Moreover, the empty set is unreachable in the subset automaton. By Proposition 2.3, all the reachable states of the subset automaton of N are pairwise distinguishable. $\square$

We did some computations, and it seems that the upper bound cannot be met in the binary case. The next theorem provides a lower bound on the state complexity of catenation on binary non-returning languages. However, our computations show that this lower bound can be exceeded.

Lemma 5.5. Let $m, n \geq 3$. There exist binary non-returning regular languages K and L with $\mathrm{sc}(K) = m$ and $\mathrm{sc}(L) = n$ such that $\mathrm{sc}(KL) \geq (m-2)2^{n-1} + 2^{n-2} + 2$.

Proof:
Let $m, n \geq 3$. Consider the languages K and L accepted by binary minimal non-returning DFAs $A = (Q_A, \{a, b\}, \delta_A, s_A, \{q_{m-2}\})$ and $B = (Q_B, \{a, b\}, \delta_B, s_B, \{n-2\})$ shown in Figure 6, in which

$Q_A = \{s_A\} \cup \{q_0, q_1, \ldots, q_{m-2}\}$,

$Q_B = \{s_B\} \cup \{0, 1, \ldots, n-2\}$,

and the transition functions δ_A and δ_B are defined as follows:

$\delta_A(s_A, a) = \delta_A(s_A, b) = q_0$,
$\delta_A(q_i, a) = q_{(i+1) \bmod (m-1)}$,
$\delta_A(q_i, b) = q_i$,

$\delta_B(s_B, a) = \delta_B(s_B, b) = 0$,
$\delta_B(j, a) = (j+1) \bmod (n-1)$,
$\delta_B(0, b) = 0$,
$\delta_B(j, b) = j + 1$, if $1 \leq j \leq n-3$,
$\delta_B(n-2, b) = 0$.

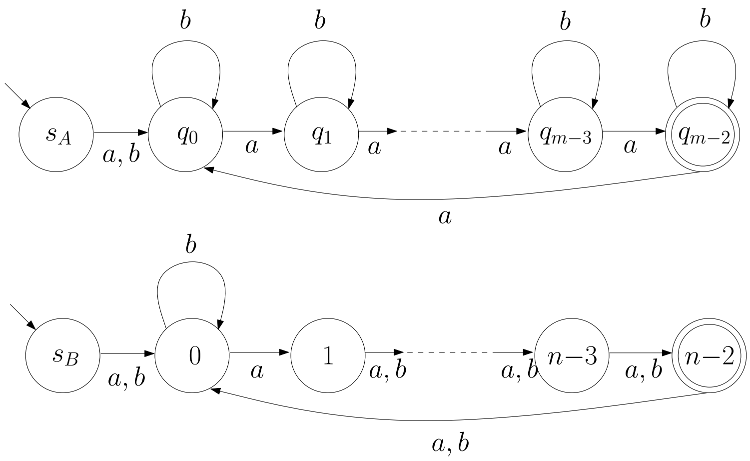

Figure 6. The DFAs of binary non-returning languages K and L with $\mathrm{sc}(KL) \geq (m-2)2^{n-1} + 2^{n-2} + 2$.

Construct an NFA N for the language KL from the two DFAs A and B by adding the transitions on a and b from the state q_{m-2} to the state 0, and by omitting the state s_B. The initial state of N is s_A and the unique final state is the state $n-2$. Let us show that the subset automaton of the NFA N has $(m-2)2^{n-1} + 2^{n-2} + 2$ reachable and pairwise distinguishable states.

We prove, by induction on the size of reachable sets, that $\{s_A\}$, $\{q_0\}$, and all sets $\{q_i\} \cup T$, where $0 \leq i \leq m-2$ and $T \subseteq \{0, 1, \ldots, n-2\}$, and such that if $i = 0$ then $0 \in T$, are reachable in the subset automaton. Each singleton set $\{q_i\}$ is reached from the initial state $\{s_A\}$ by a^{i+1}.

Let $1 \leq k \leq n-2$, and assume that the claim holds for k. Let $S = \{q_i, j_1, j_2, \ldots, j_k\}$ be a set of size $k+1$ with $0 \leq j_1 < j_2 < \cdots < j_k \leq n-2$. Consider the following six cases:

(*i*) $i = 0$ and $j_1 = 0$. Then S is reached from $\{q_{m-2}, j_2 - 1, \ldots, j_k - 1\}$ by a, and the latter set is reachable by the induction hypothesis;

(*ii*) $i = 1, j_1 = 0$ and $|S| = 2$; namely, $S = \{q_1, 0\}$. Then S is reached from $\{q_0, 0\}$ by ab^{n-2} and the latter set is reachable by (*i*);

(*iii*) $i = 1$, $j_1 = 0$, $j_2 = 1$. Then S is reached from $\{q_0, 0, j_3 - 1, \ldots, j_k - 1, n - 2\}$ by a, and the latter set is reachable by (*i*);

(*iv*) $i = 1$, $j_1 = 0$, and $j_2 \geq 2$. Then S is reached from $\{q_1, 0, 1, j_3 - j_2 + 1, \ldots, j_k - j_2 + 1\}$ by b^{j_2-1}, and the latter set is reachable by (*iii*);

(*v*) $i = 1$ and $j_1 \geq 1$. Then S is reached from $\{q_0, 0, j_2 - j_1, \ldots, j_k - j_1\}$ by ab^{j_1-1}, and the latter set is reachable by (*i*);

(*vi*) $i \geq 2$. Then S is reached from $\{q_1, (j_1 - i + 1) \bmod (n - 1), \ldots, (j_k - i + 1) \bmod (n - 1)\}$ by a^{i-1}, and the latter set is reachable by (*ii*)-(*v*).

This proves the reachability of $2 + 2^{n-2} + (m - 2)2^{n-1}$ subsets.

To prove distinguishability, let us show that we can assign a string w_q to each state q of N, except for the initial state, such that the string w_q is accepted by N only from the state q. Notice that the following strings are accepted by N only from the corresponding states:

(*i*) the string a^{n-2-j}, where $0 \leq j \leq n - 2$, is accepted by the NFA N only from the state j,

(*ii*) the string $b^n a \cdot a^{n-2}$ is accepted only from q_{m-2}, and

(*iii*) the string $a^{m-2-i} \cdot b^n a \cdot a^{n-2}$, where $0 \leq i \leq m - 3$, is accepted only from the state q_i.

The empty set is unreachable in the subset automaton of the NFA N. By Proposition 2.3, all reachable subsets of the subset automaton are pairwise distinguishable. □

Now we consider the unary case. The upper bound on the state complexity of catenation on unary regular languages is mn, and it can be met if $\gcd(m, n) = 1$ [2, Theorem 5.4]. We show that for non-returning languages, the bound $(m - 1)(n - 1) + 2$ can be met if $m - 1$ and $n - 1$ are relatively prime numbers.

Lemma 5.6. Let $m, n \geq 2$ and $\gcd(m - 1, n - 1) = 1$. There exist unary non-returning languages K and L with $\mathrm{sc}(K) = m$ and $\mathrm{sc}(L) = n$ such that $\mathrm{sc}(KL) = (m - 1)(n - 1) + 2$.

Proof:
Consider unary non-returning languages $K = a^{m-1}(a^{m-1})^*$ and $L = a^{n-1}(a^{n-1})^*$. The languages K and L are accepted by unary minimal non-returning DFAs of m and n states, respectively. The catenation of K and L is the language

$$KL = \{a^i \mid i = k(m - 1) + \ell(n - 1) \text{ and } k, \ell > 0\}.$$

The largest integer that cannot be expressed as $k(m - 1) + \ell(n - 1)$ with $k, \ell > 0$ is $(m - 1)(n - 1)$ [2, Lemma 5.1]. It follows that $\mathrm{sc}(KL) = (m - 1)(n - 1) + 2$. □

The following theorem summarizes the results of this section.

Theorem 5.7. (Catenation) Let $m, n \geq 2$. Let $f_k(m,n)$ be the state complexity of the catenation operation on non-returning languages over a k-letter alphabet defined as

$f_k(m,n) = \max\{\mathrm{sc}(KL) \mid K, L \subseteq \Sigma^*, |\Sigma| = k, \mathrm{sc}(K) = m, \mathrm{sc}(L) = n,$ and K, L non-returning$\}$.

Then

$$(i)\ \ f_k(m,n) = \begin{cases} m+1, & \text{if } m \geq 2, n = 2, \text{ and } k \geq 1; \\ 2^{n-1}+1, & \text{if } m = 2, n \geq 3, \text{ and } k \geq n-1; \\ (m-1)2^{n-1}+1, & \text{if } m, n, k \geq 3, \end{cases}$$

(*ii*) $f_k(2,n) < 2^{n-1}+1$ if $n \geq 3$ and $k < n-1$,

(*iii*) $(m-2)2^{n-1} + 2^{n-2} + 2 \leq f_2(m,n) \leq (m-1)2^{n-1}+1$ if $m, n \geq 3$,

(*iv*) $f_1(m,n) \leq mn$, and $f_1(m,n) \geq (m-1)(n-1)+2$ if $\gcd(m-1, n-1) = 1$.

Proof:
The tight bounds in (*i*) are given by Lemmas 5.1-5.4. The result in (*ii*) is proved in Lemma 5.3. The lower bound in (*iii*) is given by Lemma 5.5, and the upper bound follows from the fact that $f_2(m,n) \leq f_3(m,n)$. The upper bound in (*iv*) is the same as in the general case of regular languages, and the lower bound is given by Lemma 5.6. □

As for the binary case, our computations show that the upper bound $(m-1)2^{n-1}+1$ cannot be met, while our lower bound $(m-2)2^{n-1} + 2^{n-2} + 2$ can be exceeded.

Table 1. Tight bounds computations for catenation on binary non-returning languages.

Case	$(m-2)2^{n-1} + 2^{n-2} + 2$	$\mathrm{sc}(KL)$	$(m-1)2^{n-1}+1$
$m=3, n=3$	8	9	9
$m=3, n=4$	14	14	17
$m=4, n=4$	22	22	25
$m=4, n=5$	42	46	49
$m=5, n=5$	58	62	65

6. Reversal

The tight bound on the state complexity of the reversal operation on regular languages is 2^n with worst-case examples defined over a binary alphabet [2, 25, 26, 27]. The aim of this section is to show that for non-returning languages, the tight bound is the same. However, to prove tightness, we need a three-letter alphabet.

Lemma 6.1. Let L be a non-returning language over an alphabet Σ with $\mathrm{sc}(L) = n$, where $n \geq 3$. Then $\mathrm{sc}(L^R) \leq 2^n$, and the bound is tight if $|\Sigma| \geq 3$.

Proof:
The upper bound 2^n is the same as in the general case of regular languages. To prove tightness, consider the ternary non-returning language accepted by the DFA A shown in Figure 7, in which the transitions are defined as follows. Each state i with $2 \leq i \leq n-1$ goes to state $i-1$ on symbols a, b, and c. State 1 goes to state 0 on symbols a and c, and it goes to state $n-2$ on symbol b. The state 0 goes to itself on symbols a and b, and it does to state $n-2$ on symbol c.

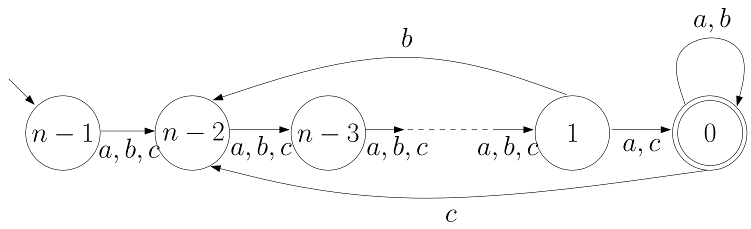

Figure 7. The ternary non-returning witness for reversal meeting the bound 2^n.

Construct the reverse A^R of the DFA A by making state 0 initial and state $n-1$ final, and by reversing all the transitions. The NFA A^R is shown in Figure 8. Let us show that the subset automaton of the NFA A^R has 2^n reachable states.

The initial state is $\{0\}$, and each singleton set $\{i\}$ with $1 \leq i \leq n-2$ is reached from $\{0\}$ by c^i. The set $\{n-1\}$ is reached from $\{n-2\}$ by a, and the empty set is reached from $\{n-1\}$ by a.

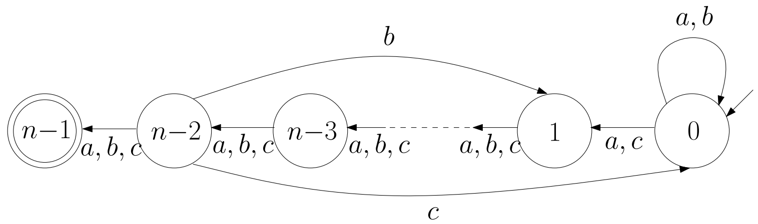

Figure 8. The reverse of the DFA from Figure 7.

Let $2 \leq k \leq n$, and assume that every subset of $\{0, 1, \dots, n-1\}$ of size $k-1$ is reachable. Let $S = \{i_1, i_2, \dots, i_k\}$ be a set of size k with $0 \leq i_1 < i_2 < \cdots < i_k \leq n-1$. Consider the following four cases:

(*i*) $i_k \leq n-2$. Then the set $\{0, i_3 - i_2, \dots, i_k - i_2\}$ is reachable by the induction hypothesis. Since we have $\{0, i_3 - i_2, \dots, i_k - i_2\} \xrightarrow{a} \{0, 1, i_3 - i_2 + 1, \dots, i_k - i_2 + 1\} \xrightarrow{b^{i_2 - i_1 - 1}}$ $\{0, i_2 - i_1, i_3 - i_1, \dots, i_k - i_1\} \xrightarrow{c^{i_1}} \{i_1, i_2, \dots, i_k\} = S$, the set S is reachable.

(*ii*) $i_k = n-1$ and $i_1 = 0$. Then S is reached from $\{i_2 - 1, \dots, i_{k-1} - 1, n-2\}$ by c, and the latter set is reachable by the induction hypothesis.

(*iii*) $i_k = n-1$ and $i_1 = 1$. Then S is reached from $\{i_2 - 1, \dots, i_{k-1} - 1, n-2\}$ by b, and the latter set is reachable by the induction hypothesis.

(*iv*) $i_k = n-1$ and $i_1 \geq 2$. Then S is reached from $\{i_1 - 1, \dots, i_{k-1} - 1, n-2\}$ by a, and the latter set is reachable by (*i*).

By Proposition 2.2, all states of the subset automaton are pairwise distinguishable. $\square$

The next result provides a lower bound in the binary case.

Lemma 6.2. Let $n \geq 3$. There exists a binary non-returning language L such that $\mathrm{sc}(L) = n$ and $\mathrm{sc}(L^R) = 2^{n-2}$.

Proof:
Let L be the binary language accepted by the minimal non-returning automaton shown in Figure 9. Then $L^R = (a+b)^*a(a+b)^{n-3}$, and it is well-known that the state complexity of L^R is 2^{n-2}. □

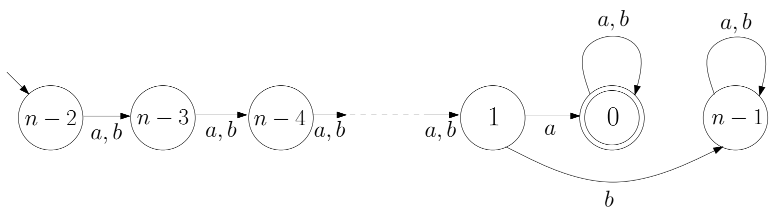

Figure 9. The DFA of a binary non-returning language L with $\mathrm{sc}(L^R) = 2^{n-2}$.

Taking into account that the reverse of every unary language is the same language, and that $\{\lambda\}^R = \{\lambda\}$ and $(\Sigma\Sigma^*)^R = \Sigma\Sigma^*$, we can summarize our results on the reversal operation on non-returning languages in the following theorem.

Theorem 6.3. **(Reversal)** Let $n \geq 3$. Let $f_k(n)$ be the state complexity of the reversal operation on non-returning regular languages over a k-letter alphabet defined as $f_k(n) = \max\{\mathrm{sc}(L^R) \mid L \subseteq \Sigma^*, |\Sigma| = k, \mathrm{sc}(L) = n, \text{ and } L \text{ is non-returning}\}$. Then

(*i*) if $k \geq 3$ then $f_k(n) = \begin{cases} 2, & \text{if } n = 2; \\ 2^n, & \text{if } n \geq 3, \end{cases}$

(*ii*) $2^{n-2} \leq f_2(n) \leq 2^n$,

(*iii*) $f_1(n) = n$.

As for the binary case, our computations again show that the upper bound 2^n cannot be met, while our lower bound 2^{n-2} can be exceeded.

7. Kleene Star

The state complexity of Kleene star on regular languages is $2^{n-1} + 2^{n-2}$ for an alphabet of at least two symbols, and it is $(n-1)^2 + 1$ in the unary case [2]. Here we show that in the case of non-returning languages over an alphabet of at least two symbols, the tight bound is 2^{n-1}. In the unary case, we get the tight bound $(n-2)^2 + 2$.

Lemma 7.1. Let L be a non-returning regular language over an alphabet Σ with $\mathrm{sc}(L) = n$, where $n \geq 2$. Then $\mathrm{sc}(L^*) \leq 2^{n-1}$, and the bound is tight if $|\Sigma| \geq 2$.

Table 2. Tight bounds computations for reversal on binary non-returning languages.

Case	2^{n-2}	$\mathrm{sc}(L^R)$	2^n
$n=3$	2	7	8
$n=4$	4	13	16
$n=5$	8	25	32
$n=6$	16	47	64
$n=7$	32	89	128

Proof:
Let $A = (Q, \Sigma, \delta, s, F)$ be a minimal non-returning automaton for L. Construct an NFA N for the language L^* from the DFA A by making the initial state s final, and by adding a transition on every symbol a from every final state to the state $\delta(s, a)$. The NFA N is non-returning, and therefore the subset automaton of N has at most $2^{n-1} + 1$ reachable states. Since A is a complete DFA, the empty set is unreachable in the subset automaton. This gives the upper bound 2^{n-1}.

To prove tightness, consider the binary language accepted by the DFA A shown in Figure 10, in which the initial state s goes to state 0 by both a and b, state 0 goes to itself by b, there is a cycle $(0, 1, \ldots, n-2)$ on symbol a, and a cycle $(1, 2, \ldots, n-1)$ on symbol b. If $n = 2$, then the final state 0 goes to itself by both a and b, so the automaton accepts the language $\{\lambda\}$.

Construct an NFA N for the language L^* from the DFA A by making the initial state s final, and by adding the transition on b from state $n-2$ to state 0.

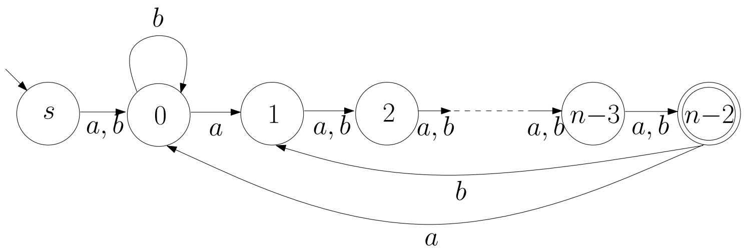

Figure 10. The binary non-returning witness for Kleene star meeting the bound 2^{n-1}.

Let us prove by induction on the size of subsets that every non-empty subset of $\{0, 1, \ldots, n-2\}$ is reachable in the subset automaton of the NFA N. Every singleton set $\{i\}$ is reached from the initial state $\{s\}$ by a^{i+1}. Let $2 \leq k \leq n-1$, and assume that every subset of size $k-1$ is reachable. Let $S = \{i_1, i_2, \ldots, i_k\}$ be a set of size k with $0 \leq i_1 < i_2 < \cdots < i_k \leq n-2$. Consider the following three cases:

(*i*) $i_1 = 0$ and $i_2 = 1$. Then S is reached from the set $\{i_3 - 1, \ldots, i_k - 1, n-2\}$ by b, and the latter set is reachable by the induction hypothesis;

(*ii*) $i_1 = 0$ and $i_2 \geq 2$. Then S is reached from $\{0, 1, i_3 - i_2 + 1, \ldots, i_k - i_2 + 1\}$ by $b^{i_2 - 1}$, and the latter set is reachable by (*i*);

(*iii*) $i_1 \geq 1$. Then S is reached from $\{0, i_2 - i_1, \ldots, i_k - i_1\}$ by a^{i_1}, and the latter set is reachable by (*i*) and (*ii*).

To prove distinguishability, notice that for each i with $0 \leq i \leq n-2$, the NFA N accepts the string a^{n-2-i} only from the state i. Moreover, the empty set is unreachable. By Proposition 2.3, all states of the subset automaton are pairwise distinguishable. □

Now we consider the unary case. The state complexity of Kleene star on unary regular languages is $(n-1)^2 + 1$ [2]. Our aim is to show that the tight bound on the state complexity of Kleene star on unary non-returning regular languages is $(n-2)^2 + 2$.

Lemma 7.2. Let L be a unary non-returning regular language with $\mathrm{sc}(L) = n$, where $n \geq 2$. Then $\mathrm{sc}(L^*) \leq (n-2)^2 + 2$, and the bound is tight.

Proof:
It has been shown by Čevorová [28, Theorems 7 and 8] that if $n \geq 6$, then in the range from $(n-2)^2 + 2$ to $(n-1)^2 + 1$, only the values $(n-2)^2 + 2$, $n^2 - 3n + 2$, $n^2 - 3n + 3$, and $(n-1)^2 + 1$ are attainable by the state complexity of the star of a unary n-state DFA language. She also proved that among these values, only $(n-2)^2 + 2$ is met by a non-returning language. The DFA of such a language is shown in Figure 11.

If $2 \leq n \leq 5$, then the direct computations show that the upper bound is $(n-2)^2 + 2$. This upper bound is met by the unary non-returning language accepted by the DFA shown in Figure 11 if $n \in \{3, 4, 5\}$, and by the language $\{\lambda\}$ if $n = 2$ [29]. □

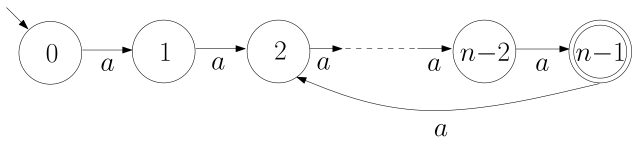

Figure 11. The unary non-returning witness for Kleene star meeting the bound $(n-2)^2 + 2$.

The next theorem summarizes our results on the Kleene star operation.

Theorem 7.3. (Kleene Star) Let $n \geq 2$. Let $f_k(n)$ be the state complexity of the star operation on non-returning regular languages over a k-letter alphabet defined as $f_k(n) = \max\{\mathrm{sc}(L^*) \mid L \subseteq \Sigma^*, |\Sigma| = k, \mathrm{sc}(L) = n,$ and L is non-returning$\}$. Then

(i) $f_k(n) = 2^{n-1}$ if $k \geq 2$,

(ii) $f_1(n) = (n-2)^2 + 2$.

8. Conclusions

The state complexity of subfamilies of regular languages, such as finite languages, unary languages, prefix-free or suffix-free regular languages, is often smaller than the state complexity of regular languages [5, 6, 7, 15, 16, 30]. We have considered another subfamily of regular languages, non-returning regular languages. Note that when a minimal DFA A is non-returning, then we say that the language $L(A)$ is non-returning.

The non-returning property is a necessary condition for a DFA to accept a suffix-free regular language, but it is not sufficient [16]. We notice that a suffix-free DFA always has a sink state whereas a non-returning DFA may not have any sink state. Based on these observations, we have examined non-returning DFAs and established the state complexities of some basic operations for non-returning regular languages.

Table 3. Comparison table between the state complexity of basic operations on suffix-free [16, 30, 31], non-returning, and general regular languages [1, 2, 22, 23, 26].

operation	suffix-free	$\|\Sigma\|$	non-returning	$\|\Sigma\|$	general	$\|\Sigma\|$
L^c	n	1	n	1	n	1
$K \cup L$	$mn-(m+n-2)$	2	$mn-(m+n-2)$	2	mn	2
$K \cap L$	$mn-2(m+n-3)$	2	$mn-(m+n-2)$	2	mn	2
$K \setminus L$	$mn-(m+2n-4)$	2	$mn-(m+n-2)$	2	mn	2
$K \oplus L$	$mn-(m+n-2)$	2	$mn-(m+n-2)$	2	mn	2
L^*	$2^{n-2}+1$	2	2^{n-1}	2	$3/4 \cdot 2^n$	2
L^R	$2^{n-2}+1$	3	2^n	3	2^n	2
$K \cdot L$	$(m-1)2^{n-2}+1$	3	$(m-1)2^{n-1}+1$	3	$(2m-1)2^{n-1}$	2

Our results are usually smaller than the state complexities for general regular languages and larger than the state complexities for suffix-free regular languages as summarized in Table 3, where we give also the size of an alphabet used for defining worst-case examples.

Table 4. The state complexity of basic operations on binary and unary non-returning regular languages.

operation	binary non-returning	unary non-returning
L^c	n	n
$K \cup L$	$mn-(m+n-2)$	$mn-(m+n-2)$ if $\gcd(m-1,n-1)=1$
$K \cap L$	$mn-(m+n-2)$	$mn-(m+n-2)$ if $\gcd(m-1,n-1)=1$
$K \setminus L$	$mn-(m+n-2)$	$mn-(m+n-2)$ if $\gcd(m-1,n-1)=1$
$K \oplus L$	$mn-(m+n-2)$	$mn-(m+n-1)$ if $\gcd(m-1,n-1)=1$
L^*	2^{n-1}	$(n-2)^2+2$
L^R	$\geq 2^{n-2}$	n
$K \cdot L$	$\geq (2m-3)2^{n-2}+2$	$\leq mn$ $\geq mn-(m+n-3)$ if $\gcd(m-1,n-1)=1$

Notice that our witnesses for reversal and catenation are defined over a three-letter alphabet. We conjecture that the upper bounds 2^n and $(m-1)2^{n-1}+1$ for reversal and catenation, respectively, cannon be met by any binary non-returning languages. However, as shown in Table 4, we are still able to get exponential lower bounds in the binary case. To get tight bounds for reversal and catenation on binary non-returning languages seems to be a very hard problem. Table 4 also summarizes our results in the unary case.

Acknowledgments

Eom and Han were supported by the Basic Science Research Program through NRF funded by MEST (2015R1D1A1A01060097), Yonsei University Future-leading Research Initiative of 2015, and Jirásková was supported by VEGA grant 2/0084/15.

References

[1] Maslov AN. Estimates of the number of states of finite automata. Soviet Mathematics Doklady. 1970; 11:1373–1375.

[2] Yu S, Zhuang Q, Salomaa K. The State Complexities of Some Basic Operations on Regular Languages. Theoretical Computer Science. 1994;125(2):315–328. doi: 10.1016/0304-3975(92)00011-F.

[3] Yu S. State Complexity of Regular Languages. Journal of Automata, Languages and Combinatorics. 2001; 6(2):221.

[4] Salomaa A, Wood D, Yu S. On the state complexity of reversals of regular languages. Theoretical Computer Science. 2004;320(2-3):315–329. doi: 10.1016/j.tcs.2004.02.032.

[5] Câmpeanu C, II KC, Salomaa K, Yu S. State Complexity of Basic Operations on Finite Languages. In: Automata Implementation, 4th International Workshop on Implementing Automata, WIA'99, Potsdam, Germany, July 17-19, Revised Papers; 1999. p.60–70.

[6] Han Y, Salomaa K. State Complexity of Union and Intersection of Finite Languages. International Journal of Foundations of Computer Science. 2008;19(3):581–595. doi: 10.1142/S0129054108005838.

[7] Pighizzini G, Shallit J. Unary Language Operations, State Complexity and Jacobsthal's Function. International Journal of Foundations of Computer Science. 2002;13(1):145–159.

[8] Holzer M, Kutrib M. Nondeterministic Descriptional Complexity of Regular Languages. International Journal of Foundations of Computer Science. 2003;14(6):1087–1102. doi: 10.1142/S0129054103002199.

[9] Câmpeanu C, Salomaa K, Yu S. Tight Lower Bound for the State Complexity of Shuffle of Regular Languages. Journal of Automata, Languages and Combinatorics. 2002;7(3):303–310.

[10] Domaratzki M. State Complexity of Proportional Removals. Journal of Automata, Languages and Combinatorics. 2002;7(4):455–468.

[11] Domaratzki M, Salomaa K. State Complexity of Shuffle on Trajectories. Journal of Automata, Languages and Combinatorics. 2004;9(2-3):217–232.

[12] Nicaud C. Average State Complexity of Operations on Unary Automata. In: Mathematical Foundations of Computer Science 1999, 24th International Symposium, MFCS'99, Szklarska Poreba, Poland, September 6-10, Proceedings; 1999. p. 231–240.

[13] Jirásková G, Okhotin A. State complexity of cyclic shift. RAIRO - Theoretical Informatics and Applications. 2008;42(2):335–360. Available from: `http://eudml.org/doc/246037`.

[14] Salomaa A, Salomaa K, Yu S. State complexity of combined operations. Theoretical Computer Science. 2007;383(2-3):140–152. doi: 10.1016/j.tcs.2007.04.015

[15] Han Y, Salomaa K, Wood D. Operational State Complexity of Prefix-Free Regular Languages. In: Automata, Formal Languages, and Related Topics - Dedicated to Ferenc Gécseg on the occasion of his 70th birthday; 2009. p. 99–115.

[16] Han Y, Salomaa K. State complexity of basic operations on suffix-free regular languages. Theoretical Computer Science. 2009;410(27-29):2537–2548. doi: 10.1016/j.tcs.2008.12.054

[17] Berstel J, Perrin D. Theory of Codes. Academic Press, Inc. Orlando; 1985. ISBN: 0120934205.

[18] Shallit JO. A Second Course in Formal Languages and Automata Theory. Cambridge University Press New York; 2008. ISBN: 0521865727, 9780521865722.

[19] Wood D. Theory of Computation. John Wiley & Sons, Inc. Boston; 1987. ISBN: 0060472081.

[20] Yu S. Regular languages. In: Rozenberg G, Salomaa A, editors. Word, Language, Grammar. vol. 1 of Handbook of Formal Languages. Springer-Verlag; 1997. p. 41–110. ISBN: 3-540-60420-0.

[21] Brzozowski JA. Derivatives of Regular Expressions. Journal of the ACM. 1964;11(4):481–494.

[22] Brzozowski JA. Quotient Complexity of Regular Languages. Journal of Automata, Languages and Combinatorics. 2010;15(1/2):71–89.

[23] Jirásková G. State complexity of some operations on binary regular languages. Theoretical Computer Science. 2005;330(2):287–298. doi: 10.1016/j.tcs.2004.04.011.

[24] Jirásková G, Palmovský M. Kleene Closure and State Complexity. In: Conference on Information Technologies - Applications and Theory, Donovaly, Slovakia, September 11-15, Proceedings; 2013. p. 94–100.

[25] Jirásková G, Sebej J. Reversal of binary regular languages. Theoretical Computer Science. 2012;449:85–92. doi: 10.1016/j.tcs.2012.05.008.

[26] Leiss EL. Succint Representation of Regular Languages by Boolean Automata. Theoretical Computer Science. 1981;13:323–330. doi: 10.1016/S0304-3975(81)80005-9.

[27] Sebej J. Reversal of regular languages and state complexity. In: Conference on Theory and Practice of Information Technologies, ITAT 2010, Hotel Smrekovica, Veľká Fatra, Slovak Republic, September 21-25, Proceedings; 2010. p. 47–54.

[28] Cevorová K. Kleene Star on Unary Regular Languages. In: Descriptional Complexity of Formal Systems - 15th International Workshop, DCFS 2013, London, ON, Canada, July 22-25, Proceedings; 2013. p. 277–288. doi: 10.1007/978-3-642-39310-5_26.

[29] Cevorová K. Kleene star on unary non-returning languages. Personal communication. 2014;.

[30] Cmorik R, Jirásková G. Basic Operations on Binary Suffix-Free Languages. In: Mathematical and Engineering Methods in Computer Science - 7th International Doctoral Workshop, MEMICS 2011, Lednice, Czech Republic, October 14-16, Revised Selected Papers; 2011. p. 94–102. doi: 10.1007/978-3-642-25929-6_9.

[31] Jirásková G, Olejár P. State Complexity of Intersection and Union of Suffix-Free Languages and Descriptional Complexity. In: Workshop on Non-Classical Models for Automata and Applications - NCMA 2009, Wroclaw, Poland, August 31 - September 1, Proceedings; 2009. p. 151–166.