State Complexity of Basic Operations on Non-returning Regular Languages

Hae-Sung Eom [1,⋆], Yo-Sub Han [1,⋆], and Galina Jirásková [2,⋆⋆]

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

[2] 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, reversal, 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.

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 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 [17] provided, without giving proofs, the state complexity of union, catenation, and star, and later Yu et al. [24] 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.

⋆ Research supported by the Basic Science Research Program through NRF funded by MEST (2012R1A1A2044562).

⋆⋆ Research supported by VEGA grant 2/0183/11 and by grant APVV-0035-10.

H. Jürgensen and R. Reis (Eds.): DCFS 2013, LNCS 8031, pp. 54–65, 2013.

Yu [25] gave a comprehensive survey of the state complexity of regular languages. Salomaa *et al.* [21] studied classes of languages, for which the reversal operation reaches the exponential upper bound. As special cases of the state complexity, researchers examined the state complexity of finite languages [4,9], the state complexity of unary language operations [19] and the nondeterministic descriptional complexity of regular languages [12]. There are several other results with respect to the state complexity of different operations [5,7,8,18].

For regular language codes, which preserve certain structural properties in the corresponding minimal DFAs, Han *et al.* [11] studied the state complexity of prefix-free regular languages. Similarly, based on suffix-freeness, Han and Salomaa [10] 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 [1]. 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 [10]. 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 observation intrigues us to investigate DFAs with non-returning property and the state complexity of basic operations on languages accepted by non-returning DFAs.

Note that state complexity of non-returning regular languages is different from the state complexity of arbitrary regular languages because there is a structural property in a non-returning DFA; the start state has no in-transitions. We get the tight bounds on the state complexity of four Boolean operations (union, intersection, difference, symmetric difference), of catenation, reversal and Kleene-star. 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 Boolean operations, catenation, reversal, and Kleene star in Sections 3, 4, 5, and 6, respectively. We summarize the state complexity results and compare them with the regular language case and the suffix-free case in Section 7.

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 reversal 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 size 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, the reader may refer to the textbooks [22,23,26]. To conclude this section let us state some preliminary results that we will use later throughout the paper.

Proposition 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 reversal. It can be easily proved using Proposition 1, and for the sake of completeness, we present the proof here.

Proposition 2 ([2]). *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 1, and therefore all states of the subset automaton of A^R are pairwise distinguishable. $\square$

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 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.* $\square$

3 Boolean Operations

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 [3,24].

In the case of non-returning languages, we obtain the precise state complexity for these operations, which again turn out to be the same. A general boolean operation with two arguments is denoted by $K \circ L$.

Theorem 1. *Let K and L be non-returning languages over an alphabet Σ with* $\mathrm{sc}(K) = m$ *and* $\mathrm{sc}(L) = n$*, where* $m, n \geq 3$*. Then* $\mathrm{sc}(K \circ L) \leq (m-1)(n-1)+1$*, and the bound is tight if* $|\Sigma| \geq 2$*.*

Proof. Let K and L be accepted by a nonreturning 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 cross-product automaton for $K \circ L$ with the state set $Q_A \times Q_B$. Since both DFAs are non-returning, in the cross-product automaton, the states (s_A, q) and (p, s_B), except for the initial state (s_A, s_B), are non-reachable. This gives the upper bound.

To prove tightness, first consider intersection. 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 Fig. 1.

In the cross-product automaton for the language $K \cap L$, the unique final state is $(m-1, n-1)$. The cross-product automaton in the case of $m = 4$ and $n = 5$ is shown in Fig. 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 cross-product automaton. If $i < k$, then the string $a^{m-1-k}b^n$ is accepted from (k,ℓ) and rejected from (i,j).

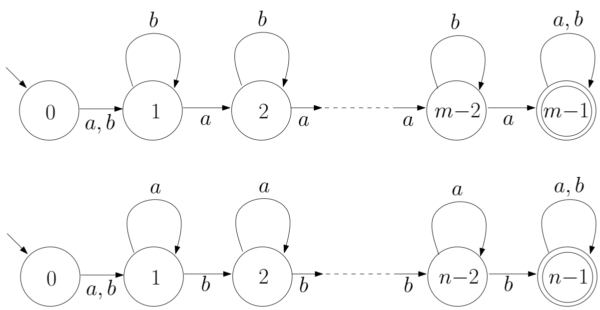

Fig. 1. The witnesses for intersection meeting the bound $(m-1)(n-1)+1$

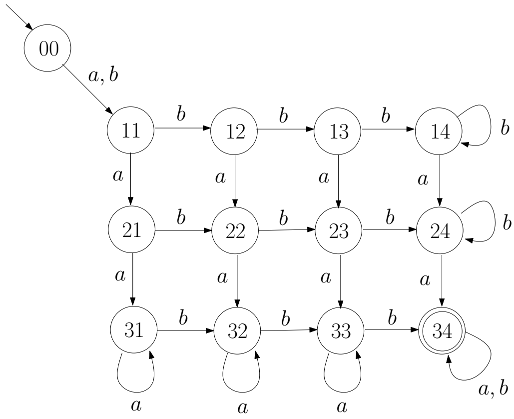

Fig. 2. The cross-product automaton for intersection; $m = 4, n = 5$

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, and concludes the proof for intersection.

To prove the tightness for union, notice that the state complexity of a regular language is the same as the state complexity of its complement. Consider the languages K^c and L^c, where K and L are the witness languages for intersection. 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 $\mathrm{sc}(K \setminus L^c) = (m-1)(n-1)+1$.

For symmetric difference, consider the same languages as for intersection. In the cross-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 of the theorem. □

4 Catenation

The state complexity of catenation on regular languages is given by the function $m2^n - 2^{n-1}$, and the worst case examples can be defined over a binary alphabet [17,24]. The next result gives the tight bound for catenation on non-returning languages over an alphabet of at least three symbols.

Theorem 2. *Let K and L be non-returning languages over an alphabet Σ with $\mathrm{sc}(K) = m$ and $\mathrm{sc}(L) = n$, where $m, n \geq 3$. Then $\mathrm{sc}(K \cdot L) \leq (m-1)2^{n-1}+1$, and the bound is tight if $|\Sigma| \geq 3$.*

Proof. To prove the upper bound, let K and L be accepted by minimal non-returning DFAs $A = (Q_A, \Sigma, \delta_A, s_A, F_A)$ and $B = (Q_B, \Sigma, \delta_B, s_B, F_B)$ of m and n states, respectively.

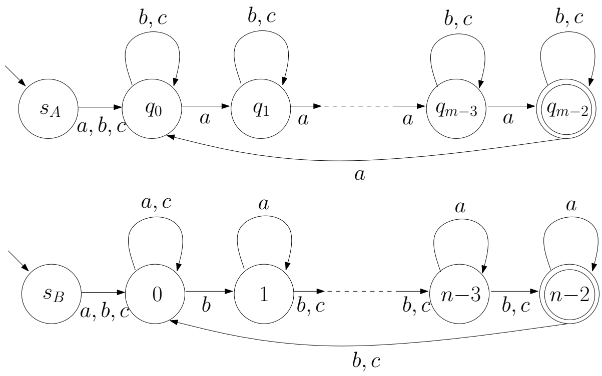

Fig. 3. The non-returning witnesses for catenation meeting the bound $(m-1)2^{n-1}+1$

Construct an NFA N for the language $K \cdot L$ 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 $\{s_A\}$, and the empty set is unreachable. It follows that the subset automaton has at most $(m-1)2^{n-1}+1$ reachable states, which proves the upper bound.

To prove tightness, let K and L be the languages accepted by the non-returning minimal DFAs A and B shown in Fig. 3.

Construct an NFA N for $K \cdot L$ from the DFAs A and B by adding transitions on a, b, c 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 $n-2$. 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 $0 \le i \le m-2$ and $0 \le j_1 < j_2 < \cdots < j_k \le n-2$, 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}. Assume that $1 \le k \le n-2$, and that the claim holds for $k-1$. Let $S = \{q_i, j_1, j_2, \ldots, j_k\}$. Consider 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 $j_1 \ge 1$. 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) $i \ge 1$. 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, notice that the NFA N accepts the string b^{n-2-j} ($0 \le j \le n-2$) only from the state j, the string $c^n b \cdot b^{n-2}$ only from the state q_{m-2}, and the string $a^{m-2-i} \cdot c^n b \cdot b^{n-2}$ $(0 \leq i \leq m-3)$ only from q_i. By Proposition 3, all states of the subset automaton of N are pairwise distinguishable. □

We did some calculations, and it seems that the upper bound cannot be met in the binary case. The next theorem provides a lower bound, however, our calculations show that it can be exceeded.

Theorem 3. *Let $m, n \geq 4$. There exist binary non-returning 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. Consider the binary languages K and L accepted by DFAs shown in Fig. 4. Construct an NFA N for the language KL from the two DFAs 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 $n-2$. 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}.

Assume that $1 \leq k \leq n-2$ and that every set S of size k and such that if $i = 0$ then $0 \in S$ is reachable. 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 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 $a \cdot b^{n-2}$ 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 the set S is reached from the set $\{q_1, 0, 1, j_3 - j_2 + 1, \ldots, j_k - j_2 + 1\}$ by $b^{j_2 - 1}$, and the latter set is reachable by (iii).

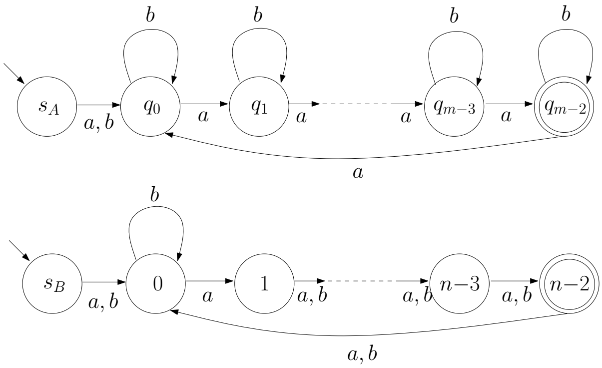

Fig. 4. Non-returning DFAs of binary K and L with $\mathrm{sc}(KL) \geq (m-2)2^{n-1} + 2^{n-2} + 2$

(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 the set S is reached from the set $\{q_1, (j_1 - i + 1) \bmod (n-1), \ldots, (j_k - i + 1) \bmod (n-1)\}$ 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, notice that the string a^{n-2-j} $(0 \leq j \leq n-2)$ is accepted by N only from the state j, the string $b^n a \cdot a^{n-2}$ only from the state q_{m-2}, and the string $a^{m-2-i} \cdot b^n a \cdot a^{n-2}$ $(0 \leq i \leq m-3)$ only from the state q_i. By Proposition 3, all subsets are pairwise distinguishable. □

5 Reversal

The tight bound on the state complexity of the reversal of regular languages is 2^n with worst-case examples defined over a binary alphabet [16,24]. 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.

Theorem 4. *Let L be a non-returning regular language over an alphabet Σ with* $\mathrm{sc}(L) = n$*, where $n \geq 4$. Then* $\mathrm{sc}(L^R) \leq 2^n$*, and the bound is tight if $|\Sigma| \geq 3$.*

Proof. The upper bound 2^n is same as in the general case.

To prove tightness, consider the language L accepted by the DFA in Fig. 5. Let us show that the subset automaton of the NFA A^R has 2^n reachable states.

The initial state of the subset automaton is $\{0\}$, and it goes by c^i to $\{i\}$ with $1 \leq i \leq n-2$. The set $\{n-2\}$ goes to $\{n-1\}$ by a. Assume that $2 \leq k \leq n$ and that every set of size $k-1$ is reachable. Let $S = \{i_1, i_2, \ldots, i_k\}$ be a set of size k with $0 \leq i_1 < i_2 < \cdots < i_k \leq n-1$. Consider four cases:

(i) $i_k \leq n-2$. Then S is reached from $\{0, i_3 - i_2, \ldots, i_k - i_2\}$ by the string $ab^{i_2-i_1-1}c^{i_1}$, and the latter set is reachable by the induction hypothesis.

(ii) $i_k = n-1$ and $i_1 = 0$. Then S is reached from $\{i_2 - 1, \ldots, i_{k-1} - 1, n-2\}$ 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 $\{i_2 - 1, \ldots, i_{k-1} - 1, n-2\}$ 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 $\{i_1 - 1, \ldots, i_{k-1} - 1, n-2\}$ by a, and the latter set is reachable by (i).

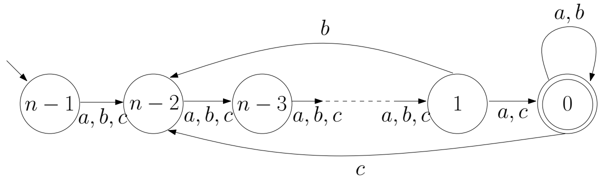

Fig. 5. The non-returning witness for reversal meeting the bound 2^n

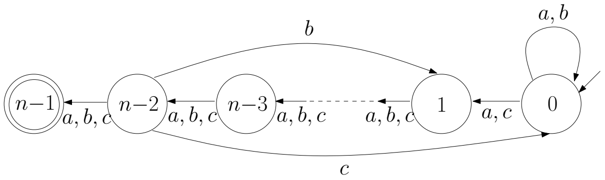

Fig. 6. The NFA A^R for the reversal of the language accepted by the DFA in Fig. 5

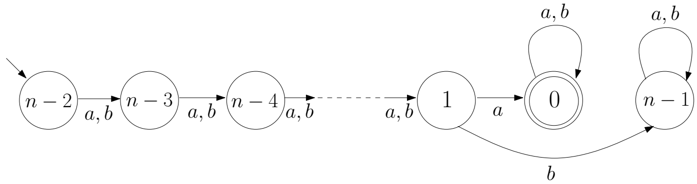

Fig. 7. The non-returning DFA of a binary language L with $\mathrm{sc}(L^R) = 2^{n-2}$

By Proposition 2, all states of the subset automaton are pairwise distinguishable, and the proof is complete. □

Our calculations show that the upper bound cannot be met by binary languages. The next result provides a lower bound in the binary case.

Theorem 5. *Let $n \geq 3$. There exists a binary non-returning regular 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 Fig. 7. Then $L^R = (a+b)^*a(a+b)^{n-3}$, and it is well-known that the state complexity of $(a+b)^*a(a+b)^{n-3}$ is 2^{n-2}. □

6 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 [24]. 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 a lower bound $(n-2)^2 + 2$, and we conjecture that this is also an upper bound.

Theorem 6. *Let L be a non-returning regular language over an alphabet Σ with $\mathrm{sc}(L) = n$, where $n \geq 3$. Then $\mathrm{sc}(L^*) \leq 2^{n-1}$, and the bound is tight if $|\Sigma| \geq 2$.*

Proof. To get an upper bound, 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 state s final, and by adding a transition on every symbol a from

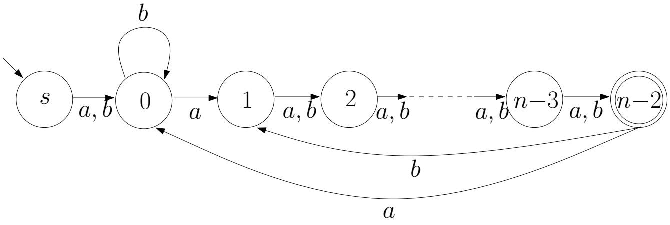

Fig. 8. The non-returning witness for Kleene star meeting the bound 2^{n-1}

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 $2^{n-1} + 1$ states. Since A is a complete DFA, the empty set is unreachable, and the upper bound is 2^{n-1}.

To prove tightness, consider the binary language accepted by the minimal n-state DFA A shown in Fig. 8. Construct an NFA N for the language L^* from the DFA A by making the state s final, and by adding the transition on b from the state $n-2$ to the state 0.

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 N. Every set $\{i\}$ is reached from the initial state $\{s\}$ by a^{i+1}. Assume that $2 \le k \le n-1$ and 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 $0 \le i_1 < i_2 < \cdots < i_k \le n-2$. Consider three cases:

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

(ii) $i_1 = 0$ and $i_2 \ge 2$. 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) $i_1 \ge 1$. 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 the NFA N accepts the string a^{n-2-i}, where $0 \le i \le n-2$, only from the state i. Since the empty set is unreachable in the subset automaton, by Proposition 3, all states of the subset automaton are pairwise distinguishable. □

Theorem 7. *Let $n \ge 3$. There exists a unary non-returning regular language with* $\mathrm{sc}(L) = n$ *and* $\mathrm{sc}(L^*) = (n-2)^2 + 2$.

Proof. Let L be the language accepted by the unary non-returning DFA shown in Fig. 9. Then $L^* = \{\lambda\} \cup \{a^m \mid m = x(n-1) + y(n-2), x > 0, y \ge 0\}$.

Since $gcd(n-1, n-2) = 1$, the largest integer that cannot be expressed as $x(n-1) + y(n-2)$ with $x > 0, y \ge 0$ is $(n-2)(n-2)$ [24]. It follows that the minimal DFA for L^* has $(n-2)^2 + 2$ states. □

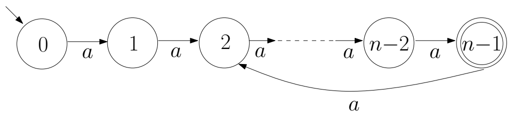

Fig. 9. The non-returning DFA of a unary language L with $\mathrm{sc}(L^*) = (n-2)^2 + 2$

7 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 [4,9,10,11,19]. 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 [10]. 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. 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 Fig. 10.

operation	non-returning	suffix-free	general
$K \cup L$	$mn-(m+n)+2$	$mn-(m+n)+2$	mn
$K \cap L$	$mn-(m+n)+2$	$mn-2(m+n)+6$	mn
$K \setminus L$	$mn-(m+n)+2$	$mn-(m+2n-4)$	mn
$K \oplus L$	$mn-(m+n)+2$	$mn-(m+n-2)$	mn
L^R	2^n	$2^{n-2}+1$	2^n
$K \cdot L$	$(m-1)2^{n-2}+1$	$(m-1)2^{n-2}+1$	$m2^n - 2^{n-1}$
L^*	2^{n-1}	$2^{n-2}+1$	$2^{n-1}+2^{n-2}$

Fig. 10. Comparison table between the state complexity of basic operations for non-returning, suffix-free, and general regular languages

For the reversal and catenation case, we use a three-letter alphabet for the lower bounds that meet the upper bounds. We conjecture that a ternary alphabet is necessary. Tight bounds for reversal and catenation in the binary case remain open. The calculations show that our lower bounds can be exceeded.

Acknowledgements. We wish to thank the referees for the careful reading of the paper and valuable suggestions.

References

1. Berstel, J., Perrin, D.: Theory of code. Academic Press, Inc. (1985)
2. Brzozowski, J.: Derivatives of regular expressions. J. ACM 11, 481–494 (1964)

3. Brzozowski, J.: Quotient complexity of regular languages. In: DCFS, pp. 17–28 (2009)
4. Câmpeanu, C., Culik II, K., Salomaa, K., Yu, S.: State complexity of basic operations on finite languages. In: Boldt, O., Jürgensen, H. (eds.) WIA 1999. LNCS, vol. 2214, pp. 60–70. Springer, Heidelberg (2001)
5. Câmpeanu, C., Salomaa, K., Yu, S.: Tight lower bound for the state complexity of shuffle of regular languages. J. Autom. Lang. Comb. 7, 303–310 (2002)
6. Cmorik, R., Jirásková, G.: Basic operations on binary suffix-free languages. In: Kotásek, Z., Bouda, J., Černá, I., Sekanina, L., Vojnar, T., Antoš, D. (eds.) MEMICS 2011. LNCS, vol. 7119, pp. 94–102. Springer, Heidelberg (2012)
7. Domaratzki, M.: State complexity of proportional removals. J. Autom. Lang. Comb. 7, 455–468 (2002)
8. Domaratzki, M., Salomaa, K.: State complexity of shuffle on trajectories. J. Autom. Lang. Comb. 9, 217–232 (2004)
9. Han, Y.-S., Salomaa, K.: State complexity of union and intersection of finite languages. Internat. J. Found. Comput. Sci. 19, 581–595 (2008)
10. Han, Y.-S., Salomaa, K.: State complexity of basic operations on suffix-free regular languages. Theoret. Comput. Sci. 410, 2537–2548 (2009)
11. Han, Y.-S., Salomaa, K., Wood. D.: Operational state complexity of prefix-free regular languages. In: Automata, Formal Languages, and Related Topics, pp. 99–115 (2009)
12. Holzer, M., Kutrib, M.: Nondeterministic descriptional complexity of regular languages. Internat. J. Found. Comput. Sci. 14, 1087–1102 (2003)
13. Jirásková, G., Šebej, J.: Reversal of binary regular languages. Theoret. Comput. Sci. 449, 85–92 (2012)
14. Jirásková, G.: State complexity of some operations on binary regular languages. Theoret. Comput. Sci. 330, 287–298 (2005)
15. Jirásková, G., Okhotin, A.: On the state complexity of star of union and star of intersection. Fund. Inform. 109, 161–178 (2011)
16. Leiss, E.: Succinct representation of regular languages by boolean automata. Theoret. Comput. Sci. 13, 323–330 (1981)
17. Maslov, A.N.: Estimates of the number of states of finite automata. Soviet Math. Dokl. 11, 1373–1375 (1970)
18. Nicaud, C.: Average state complexity of operations on unary automata. In: Kutyłowski, M., Wierzbicki, T., Pacholski, L. (eds.) MFCS 1999. LNCS, vol. 1672, pp. 231–240. Springer, Heidelberg (1999)
19. Pighizzini, G., Shallit, J.: Unary language operations, state complexity and Jacobsthal's function. Internat. J. Found. Comput. Sci. 13, 145–159 (2002)
20. Salomaa, A., Salomaa, K., Yu, S.: State complexity of combined operations. Theoret. Comput. Sci. 383, 140–152 (2007)
21. Salomaa, A., Wood, D., Yu, S.: On the state complexity of reversals of regular languages. Theoret. Comput. Sci. 320, 315–329 (2004)
22. Shallit, J.: A second course in formal languages and automata theory. Cambridge University Press, New York (2008)
23. Wood, D.: Theory of computation. John Wiley & Sons, Inc., New York (1987)
24. Yu, S., Zhuang, Q., Salomaa, K.: The state complexities of some basic operations on regular languages. Theoret. Comput. Sci. 125, 315–328 (1994)
25. Yu, S.: State complexity of regular languages. J. Autom. Lang. Comb. 6, 221–234 (2001)
26. Yu, S.: Regular languages. In: Rozenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages, vol. I, pp. 41–110 (1997)