Complement on Prefix-Free, Suffix-Free, and Non-Returning NFA Languages

Galina Jirásková* and Peter Mlynárčik

Mathematical Institute, Slovak Academy of Sciences
Grešákova 6, 040 01 Košice, Slovakia
jiraskov@saske.sk, mlynarcik1972@gmail.com

Abstract. We prove that the tight bound on the nondeterministic state complexity of complementation on prefix-free and suffix-free languages is 2^{n-1}. To prove tightness, we use a ternary alphabet, and we show that this bound cannot be met by any binary prefix-free language. On non-returning languages, the upper bound is $2^{n-1}+1$, and it is tight already in the binary case. We also study the unary case in all three classes.

1 Introduction

The complement of a formal language L over an alphabet Σ is the language $L^c = \Sigma^* \setminus L$, where Σ^* is the set of all strings over an alphabet Σ. The complementation is an easy operation on regular languages represented by deterministic finite automata (DFAs) since to get a DFA for the complement of a regular language, it is enough to interchange the final and non-final states in a DFA for this language.

On the other hand, complementation on regular languages represented by nondeterministic finite automata (NFAs) is an expensive task. First, we have to apply the subset construction to a given NFA, and only after that, we may interchange the final and non-final states. This gives an upper bound 2^n.

Sakoda and Sipser [17] gave an example of languages over a growing alphabet size meeting this upper bound on the nondeterministic state complexity of complementation. Birget claimed the result for a three-letter alphabet [3], but later corrected this to a four-letter alphabet. Ellul [7] gave binary $O(n)$-state witness languages. Holzer and Kutrib [12] proved the lower bound 2^{n-2} for a binary n-state NFA language. Finally, a binary n-state NFA language meeting the upper bound 2^n was described by Jirásková in [15]. In the unary case, the complexity of complementation is known to be in $e^{\Theta(\sqrt{n \ln n})}$ [12, 14].

In this paper, we investigate the complementation operation on prefix-free, suffix-free, and non-returning languages. A language is prefix-free if it does not contain two distinct strings one of which is a prefix of the other. The suffix-free languages are defined in a similar way. We call a language non-returning if a minimal NFA for this language does not have any transitions going to the initial state.

* Research supported by grant APVV-0035-10.

H. Jürgensen et al. (Eds.): DCFS 2014, LNCS 8614, pp. 222–233, 2014.

Prefix-free languages are used in coding theory. In prefix codes, like variable-length Huffman codes or country calling codes, there is no codeword that is a proper prefix of any other codeword. With such a code, a receiver can identify each codeword without any special marker between words.

The complexity of basic regular operations on prefix-free and suffix-free languages, both in the deterministic and nondeterministic cases, was studied by Han *at al.* in [8–11]. For the nondeterministic state complexity of complementation, they obtained an upper bound $2^{n-1}+1$ in both classes, and lower bounds 2^{n-1} and $2^{n-1}-1$ for prefix-free and suffix-free languages, respectively. The question of a tight bound remained open. In this paper, we solve this open question, and prove that in both classes, the tight bound is 2^{n-1}. To prove tightness, we use a ternary alphabet, and in the case of prefix-free languages, we show that this bound cannot be met by any binary language.

Eom *et al.* in [6] investigated also the class of so called non-returning regular languages, the minimal DFA for which has no transitions going to the initial state. It is known that every suffix-free language is non-returning, but the converse does not hold. Here we study the complementation on so called non-returning NFA languages, defined as languages represented by a minimal non-returning NFA. We show that the upper bound on the complexity of complementation in this class is $2^{n-1}+1$, and we prove that it is tight already in the binary case.

We also study the unary case, and prove that the nondeterministic state complexity of complementation is in $\Theta(\sqrt{n})$ in the class of prefix-free or suffix-free languages, and it is in $2^{\Theta(\sqrt{n\log n})}$ in the class of non-returning NFA languages.

To prove the minimality of nondeterministic finite automata, we use a fooling set lower-bound technique [1, 3, 5, 13].

Definition 1. *A set of pairs of strings* $\{(x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n)\}$ *is called a* fooling set *for a language* L *if for all* i,j *in* $\{1,2,\ldots,n\}$,

(F1) $x_iy_i \in L$, *and*

(F2) *if* $i \neq j$, *then* $x_iy_j \notin L$ *or* $x_jy_i \notin L$.

Lemma 1 ([3, 5, 13]). *Let* $\mathcal{F}$ *be a fooling set for a language* L. *Then every NFA (with multiple initial states) for the language* L *has at least* $|\mathcal{F}|$ *states.* □

2 Complement on Prefix-Free Languages

Let us start with complementation on prefix-free languages. The following two observations are easy to prove.

Proposition 1 ([9]). *Let* $n \geq 2$ *and* $A=(Q,\Sigma,\delta,s,F)$ *be a minimal* n*-state DFA for a language* L. *Then* L *is prefix-free if and only if* A *has a dead state* q_d *and exactly one final state* q_f *such that* $\delta(q_f,a)=q_d$ *for each* a *in* Σ. □

Proposition 2 ([10]). *Let* $N=(Q,\Sigma,\delta,s,F)$ *be a minimal NFA for a prefix-free language. Then* N *has exactly one final state* q_f, *and* $\delta(q_f,a)=\emptyset$ *for each* a *in* Σ. □

Han *et al.* in [10] obtained an upper bound $2^{n-1}+1$ and a lower bound 2^{n-1} on the nondeterministic complexity of complementation on prefix-free languages. Our first result shows that the upper bound can be decreased by one. Recall that the nondeterministic state complexity of a regular language L, $\text{nsc}(L)$, is defined as the smallest number of states in any NFA recognizing the language L.

Lemma 2. *Let* $n \geq 3$. *Let* L *be a prefix-free regular language with* $\text{nsc}(L) = n$. *Then* $\text{nsc}(L^c) \leq 2^{n-1}$.

Proof. Let N be an n-state NFA for a prefix-free language L. Construct the subset automaton of the NFA N and minimize it. Then, all the final states are equivalent, and they go to the dead state on each input. Thus L is accepted by a DFA $A = (Q, \Sigma, \delta, s, \{q_f\})$ with at most $2^{n-1}+1$ states, with a dead state q_d which goes to itself on each symbol, and one final state q_f which goes to the dead state on each symbol, thus $\delta(q_d, a) = q_d$ and $\delta(q_f, a) = q_d$ for each a in Σ.

To get a DFA for the language L^c, we interchange the final and non-final states in the DFA A, thus L^c is accepted by the $(2^{n-1}+1)$-state DFA $A^c = (Q, \Sigma, \delta, s, Q \setminus \{q_f\})$. We show that using nondeterminism, we can save one state, that is, we describe a 2^{n-1}-state NFA for the language L^c.

Construct a 2^{n-1}-state NFA N^c for L^c from the DFA A^c by omitting state q_d, and by replacing each transition (q, a, q_d) by two transitions (q, a, q_f) and (q, a, s). Formally, construct an NFA $N^c = (Q \setminus \{q_d\}, \Sigma, \delta', s, Q \setminus \{q_f, q_d\})$, where

$$\delta'(q,a) = \begin{cases} \{\delta(q,a)\}, & \text{if } \delta(q,a) \neq q_d, \\ \{q_f, s\}, & \text{if } \delta(q,a) = q_d. \end{cases}$$

Let us show that $L(N^c) = L(A^c)$.

Let $w = a_1 a_2 \cdots a_k$ be a string in $L(A^c)$, and let $s, q_1, q_2, \ldots, q_k$ be the computation of the DFA A^c on the string w. If $q_k \neq q_d$, then each q_i is different from q_d since q_d goes to itself on each symbol. It follows that $s, q_1, q_2, \ldots, q_k$ is also a computation of the NFA N^c on the string w. Now assume that $q_k = q_d$. Then there exists an ℓ such that the states $q_\ell, q_{\ell+1}, \ldots, q_k$ are equal to q_d, and the states $s, q_1, \ldots, q_{\ell-1}$ are not equal to q_d. If $\ell = k$, then $\delta(q_{k-1}, a_k) = q_d$, so $s \in \delta'(q_{k-1}, a_k)$. It follows that $s, q_1, q_2, \ldots, q_{k-1}, s$ is an accepting computation of N^c on w. If $\ell < k$, then we have $q_\ell = q_{\ell+1} = \cdots = q_k = q_d$, and therefore the string w is accepted in N^c through the accepting computation $s, q_1, \ldots, q_{\ell-1}, q_f, q_f, \ldots, q_f, s$ since we have $\delta'(q_{\ell-1}, a_\ell) = \{q_f, s\}$, and $\delta'(q_f, a) = \{q_f, s\}$ for each a in Σ.

Now assume that a string $w = a_1 a_2 \cdots a_k$ is rejected by the DFA A^c. Let $s = q_0, q_1, q_2, \ldots, q_k$ be the rejecting computation of the DFA A^c on the string w. Since the only non-final state of the DFA A^c is q_f, we must have $q_k = q_f$. It follows that each state q_i is different from q_d, and therefore in the NFA N^c, we have $\delta'(q_{i-1}, a_i) = \{\delta(q_{i-1}, a_i)\}$. This means that $s = q_0, q_1, q_2, \ldots, q_k$ is a unique computation of N^c on w. Since this computation is rejecting, the string w is rejected by the NFA N^c. □

To prove tightness, we use the same languages as in [10]. We provide a simple alternative proof, in which we use a fooling-set lower bound technique.

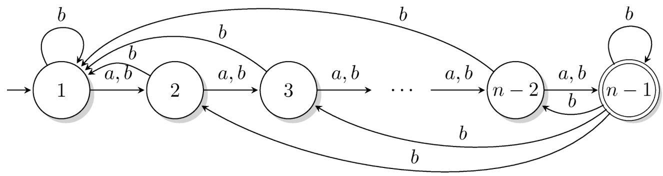

Fig. 1. An NFA of a binary regular language K with $\text{nsc}(K^c) = 2^{n-1}$

Lemma 3. *Let $n \geq 3$. There exists a ternary prefix-free language such that* $\text{nsc}(L) = n$ *and* $\text{nsc}(L^c) \geq 2^{n-1}$.

Proof. Let K be the language accepted by the NFA over $\{a, b\}$ shown in Fig. 1 with $n-1$ states. Set $L = K \cdot c$. Then L is a prefix-free language recognized by an n-state NFA in Fig. 2. As shown in [15, Theorem 5], there exists a fooling set $\mathcal{F} = \{(x_S, y_S) \mid S \subseteq \{1, 2, \ldots, n-1\}\}$ of size 2^{n-1} for the language K^c. Then the set of pairs of strings $\mathcal{F}' = \{(x_S, y_S \cdot c) \mid S \subseteq \{1, 2, \ldots, n-1\}\}$ is a fooling set of size 2^{n-1} for the language L^c. Hence, by Lemma 1, every NFA for the language L^c requires at least 2^{n-1} states. □

We summarize the results given in Lemma 2 and Lemma 3 in the following theorem which provides the tight bound on the nondeterministic state complexity of complementation on prefix-free languages. This solves an open problem from [10].

Theorem 1 (Complement on Prefix-Free Languages, $|\Sigma| \geq 3$). *Let $n \geq 3$. Let L be a prefix-free regular language over an alphabet Σ with* $\text{nsc}(L) = n$. *Then* $\text{nsc}(L^c) \leq 2^{n-1}$, *and the bound is tight if* $|\Sigma| \geq 3$. □

Notice that the bound 2^{n-1} is tight for an alphabet with at least three symbols. In Section 6, we prove that this bound cannot be met by any binary prefix-free language.

3 Complement on Suffix-Free Languages

In this section, we study the complementation operation on suffix-free languages. We first recall some definitions and known facts.

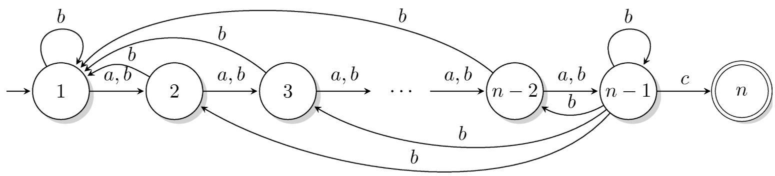

Fig. 2. An NFA of a ternary prefix-free language L with $\text{nsc}(L^c) = 2^{n-1}$

An automaton $A = (Q, \Sigma, \delta, s, F)$ is *non-returning* if the initial state s has no in-transitions, that is, for each state q and each symbol a, we have $s \notin \delta(q, a)$.

Proposition 3 ([8, 11]). *Every minimal DFA (NFA) for a non-empty suffix-free language is non-returning.*

Proposition 4. *Let $A = (Q, \Sigma, \delta, s, F)$ be a minimal DFA for a non-empty suffix-free regular language. Then A has a dead state d. Moreover, for each symbol a in Σ, there is a state q_a with $q_a \neq d$ such that $\delta(q_a, a) = d$.*

Proof. Let $a \in \Sigma$. Consider the string a^m with $m \geq |Q|$. We must have $\delta(s, a^m) = d$, where d is a dead state, because otherwise, the DFA A would accept strings $a^m w$ and $a^\ell w$ with $\ell < m$, which would be a contradiction with suffix-freeness of $L(A)$. Since $s \neq d$, there is a state q_a with $q_a \neq d$ such that $\delta(q_a, a) = d$. □

Han and Salomaa in [11] have obtained an upper bound $2^{n-1} + 1$ on the nondeterministic state complexity of complementation on suffix-free languages. Our next result shows that this upper bound can be again decreased by one.

Lemma 4. *Let $n \geq 3$. Let L be a suffix-free regular language with* $\text{nsc}(L) = n$. *Then* $\text{nsc}(L^c) \leq 2^{n-1}$.

Proof. Let N be a non-returning n-state NFA for a suffix-free language L. The subset automaton $A = (Q, \Sigma, \delta, s, F)$ of the NFA N has at most $1 + 2^{n-1}$ reachable states since the only reachable subset that contains the initial state of N is the initial state of the subset automaton. The initial state of the subset automaton is non-final since L does not contain the empty string.

After interchanging the final and non-final states, we get a DFA $A^c = (Q, \Sigma, \delta, s, Q \setminus F)$ for L^c of $1 + 2^{n-1}$ states. The initial state of A^c is final and has no in-transitions. The state d is final as well, and it accepts every string.

Construct a 2^{n-1}-state NFA N^c from the DFA A^c as follows. Let Q_d be the set of states of A^c different from d and such that they have a transition to the state d, that is, $Q_d = \{q \in Q \setminus \{d\} \mid \text{there is an } a \text{ in } \Sigma \text{ such that } \delta(q, a) = d\}$; remind that by Proposition 4, for each symbol a, there is a state q_a in Q_d that goes to d by a. Replace each transition (q, a, d) by transitions (q, a, p) for each p in Q_d, and moreover add the transition (q, a, s). Then, remove the state d. Formally, let $N^c = (Q \setminus \{d\}, \Sigma, \delta', s, (Q \setminus \{d\}) \setminus F)$, where

$$\delta'(q, a) = \begin{cases} \{\delta(q, a)\}, & \text{if } \delta(q, a) \neq d, \\ \{s\} \cup Q_d, & \text{if } \delta(q, a) = d. \end{cases}$$

In a similar way as in the case of prefix-free languages, it can be shown that $L(N^c) = L(A^c)$. □

As for a lower bound, Han and Salomaa in [11] claimed that there exists a ternary suffix-free language meeting the bound $2^{n-1} - 1$. In the next lemma, we increase this lower bound by one.

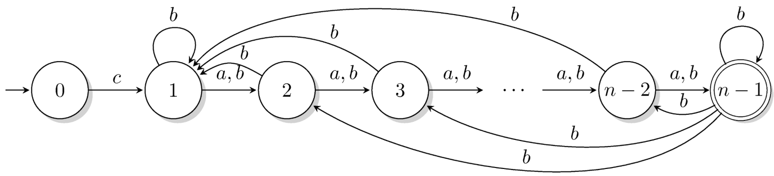

Fig. 3. An NFA of a ternary suffix-free language L with $\mathrm{sc}(L^c) = 2^{n-1}$

Lemma 5. *Let $n \geq 3$. There exists a ternary suffix-free language such that* $\mathrm{nsc}(L) = n$ *and* $\mathrm{nsc}(L^c) \geq 2^{n-1}$.

Proof. Let K be the language accepted by the NFA over $\{a, b\}$ shown in Fig. 1 with $n-1$ states. Set $L = c \cdot K$. Then L is a suffix-free language recognized by an n-state NFA shown in Fig 3. As shown in [15, Theorem 5], there exists a fooling set $\mathcal{F} = \{(x_S, y_S) \mid S \subseteq \{1, 2, \ldots, n-1\}\}$ of size 2^{n-1} for the language K^c. Then the set of pairs of strings $\mathcal{F}' = \{(c \cdot x_S, y_S) \mid S \subseteq \{1, 2, \ldots, n-1\}\}$ is a fooling set of size 2^{n-1} for the language L^c. □

We can summarize the results of this section in the following theorem which provides the tight bound on the nondeterministic state complexity of complementation on suffix-free languages over an alphabet with at least three symbols. Whether or not this bound can be met by binary languages remains open.

Theorem 2 (Complement on Suffix-Free Languages, $|\Sigma| \geq 3$). *Let $n \geq 3$. Let L be a suffix-free regular language over an alphabet Σ with* $\mathrm{nsc}(L) = n$. *Then* $\mathrm{nsc}(L^c) \leq 2^{n-1}$, *and the bound is tight if $|\Sigma| \geq 3$.* □

4 Complement on Non-Returning Languages

In this section, we consider languages that are recognized by non-returning NFAs. We call a regular language *non-returning* if it is accepted by a minimal non-returning NFA. Notice that every suffix-free language is non-returning, but the converse does not hold.

The state complexity of basic regular operations on languages represented by non-returning DFAs has been investigated by Eom *et al* in [6].

Here we study the nondeterministic state complexity of complementation on non-returning NFA languages. Our next theorem shows that in this case, the tight bound is $2^{n-1}+1$. Moreover, this bound is tight already in the binary case.

Theorem 3 (Complement on Non-Returning Languages, $|\Sigma| \geq 2$). *Let $n \geq 3$. Let L be a non-returning language over an alphabet Σ with* $\mathrm{nsc}(L) = n$. *Then* $\mathrm{nsc}(L^c) \leq 2^{n-1} + 1$, *and the bound is tight if $|\Sigma| \geq 2$.*

Proof. Let $N = (Q, \Sigma, \delta, s, F)$ be an n-state non-returning NFA for a language L. In the subset automaton of the NFA N, no subset containing the initial state s is reachable, except for the initial subset $\{s\}$. Therefore, the subset automaton has at most $2^{n-1}+1$ reachable subsets. After interchanging the final and non-final states in the subset automaton, we get a $(2^{n-1}+1)$-state DFA for L^c. This gives the upper bound.

To prove tightness, consider a non-returning language $L = b \cdot K$, where K is the language accepted by the NFA shown in Fig. 1. The n-state NFA N for the language L is shown in Fig. 4.

Let $\mathcal{F} = \{(x_S, y_S) \mid S \subseteq \{1, 2, \ldots, n-1\}\}$ be the fooling set for the language K^c described in [15, Theorem 5]; notice that x_S is a string, by which the initial state 1 of the NFA in Fig. 1 goes to the set S. Let us show that the set

$$\mathcal{F}' = \{(\varepsilon, b^{n-2}), (a, b^n)\} \cup \{(bx_S, y_S) \mid S \subseteq \{1, 2, \ldots, n-1\} \text{ and } S \neq \emptyset\}$$

is a fooling set for the language L^c.

(F1) The strings b^{n-2} and ab^n are rejected by N, so they are in L^c. Each string $x_S y_S$ is in K^c, which means that the string $bx_S y_S$ is in L^c.

(F2) If S and T are distinct and non-empty subset of $\{1, 2, \ldots, n-1\}$, then at least one of the strings $x_S y_T$ and $x_T y_S$ is in K, so at least one of $bx_S y_T$ and $bx_T y_S$ is in L, so it is not in L^c. Let S be a non-empty set of $\{1, 2, \ldots, n-1\}$. The initial state 0 goes to the set S by $b \cdot x_S$. Since S is non-empty, both strings b^{n-2} and b^n are accepted from S since they are accepted from each state in $\{1, 2, \ldots, n-1\}$. It follows that the NFA N accepts the strings $bx_S \cdot b^n$ and $bx_S \cdot b^{n-2}$, so these strings are not in L^c. Finally, the string $\varepsilon \cdot b^n$ is accepted by the NFA N, so it is not in L^c.

Hence $\mathcal{F}'$ is a fooling set for the language L^c of size $2^{n-1}+1$. By Lemma 1, every NFA for the language L^c requires at least $2^{n-1}+1$. □

5 Unary Alphabet

In this section, we consider the complementation operation on unary prefix-free, suffix-free, and non-returning languages. Our aim is to show that while in the case of prefix-free and suffix-free unary languages, the nondeterministic state complexity of complementation is in $\Theta(\sqrt{n})$, in the case of non-retuning unary languages, it is in $2^{\Theta(\sqrt{n \log n})}$. Let us start with the following observation.

Lemma 6. *Let $n \geq 3$ and $L = \{a\}^* \setminus \{a^n\}$. Then $\sqrt{n/3} \leq \operatorname{nsc}(L) \leq 6\sqrt{n}$.*

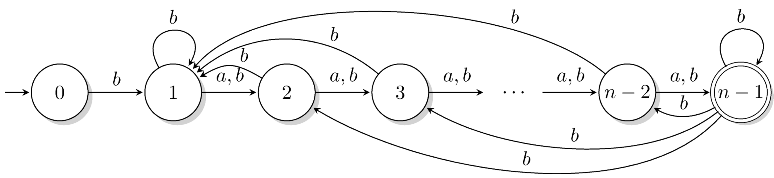

Fig. 4. An NFA of a binary non-returning language L with $\operatorname{sc}(L^c) = 2^{n-1} + 1$

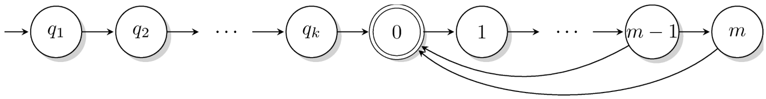

Fig. 5. An NFA A that does not accept a^n and accepts all the longer strings; $m = \lfloor\sqrt{n}\rfloor$, $k = n - (m^2 - m - 1) \leq 3\sqrt{n}$

Proof. First consider a lower bound, and let us show that every NFA for L requires at least $\sqrt{n/3}$ states. Assume for a contradiction that there is an NFA N for L with less than $\sqrt{n/3}$ states. Then the tail in the Chrobak normal form of N is of size less that $3 \cdot (\sqrt{n/3})^2$ [4, 19], thus less than n. Since a^n must be rejected, each cycle in the Chrobak normal form must contain a rejecting state. It follows that infinitely many strings are rejected, which is a contradiction.

Now let us prove the upper bound. Let $m = \lfloor\sqrt{n}\rfloor$, and consider relatively prime numbers m and $m+1$. It is known that the maximal integer that cannot be expressed as $xm + y(m+1)$ for non-negative integers x and y is $(m-1)m-1 = m^2 - m - 1$ [21]. Let $k = n - (m^2 - m - 1)$. Then $0 < k \leq 3\sqrt{n}$. Next, the NFA A shown in Fig. 5 and consisting of a path of length k and two overlapping cycles of lengths m and $m+1$ does not accept a^n, and accepts all strings a^i with $i \geq n+1$.

It remains to accept the shorter strings. To this aim let $p_1, p_2, \ldots, p_\ell$ be the first ℓ primes such that $p_1 p_2 \cdots p_\ell > n$. Then $\ell \leq \lceil \log n \rceil$. Thus $p_1 + p_2 + \cdots + p_\ell = \Theta(\ell^2 \ln \ell) \leq \sqrt{n}$ [2]. Consider an NFA B consisting of an initial state s that is connected to ℓ cycles of lengths $p_1, p_2, \ldots, p_\ell$. Let the states in the $j-th$ cycle be $0, 1, \ldots, p_j - 1$, where s is connected to state 1. The state $n \bmod p_j$ is non-final, and all the other states are final. Then this NFA does not accept a^n, but accepts all strings a^i with $i \leq n-1$ since we have $(i \bmod p_1, i \bmod p_2, \ldots, i \bmod p_\ell) \neq (n \bmod p_1, n \bmod p_2, \ldots, n \bmod p_\ell)$. The NFA B for $n = 24$ is shown in Fig. 6.

Now we get the resulting NFA for the language L of at most $6\sqrt{n}$ states as the union of NFAs A and B. □

Using the above result, we get that the nondeterministic state complexity of complementation on unary prefix-free or suffix-free languages is in $\Theta(\sqrt{n})$.

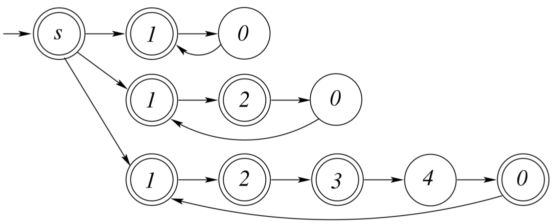

Fig. 6. The NFA B; $n = 24$

Theorem 4 (Complement on Unary Prefix- and Suffix-Free Languages). *Let L be a unary prefix-free or suffix-free regular language with* $\text{nsc}(L) = n$. *Then* $\text{nsc}(L^c) = \Theta(\sqrt{n})$.

Proof. The only prefix-free or suffix-free unary language with $\text{nsc}(L) = n$ is the singleton language $\{a^{n-1}\}$. Its complement is $\{a\}^* \setminus \{a^{n-1}\}$, and the theorem follows from Lemma 6. □

Now, we turn our attention to unary non-returning NFA languages.

In the NFA-to-DFA conversion of unary languages, a crucial role is played by the function $F(n) = \max\{\text{lcm}(x_1, \ldots, x_k) \mid x_1 + \cdots + x_k = n\}$. It is known that $F(n) \in e^{\Theta(\sqrt{n \ln n})}$ and that $O(F(n))$ states suffice to simulate an n-state NFA by a DFA [4]. This means that $O(F(n))$ states are sufficient for an NFA to accept the complement of a unary NFA language. Moreover, in [12] a unary n-state NFA language is described such that every NFA accepting its complement needs at least $F(n-1)$ states. In [14], using a fooling set method, the lower bound $F(n-1)+1$ is proved for a non-returning language. For the sake of completeness, we recall this proof here.

Lemma 7. *Let $n \geq 3$. There exits a unary n-state non-returning NFA N such that every NFA for the complement of $L(N)$ requires at least $F(n-1)+1$ states.*

Proof. Let $i_1, i_2, \ldots, i_k$ be the integers, for which the maximum in the definition of $F(n-1)$ is attained. Consider an n-state NFA N shown in Fig. 7. The NFA N consists of the initial state s and k disjoint cycles of lengths $i_1, i_2, \ldots, i_k$. The initial and rejecting state s is nondeterministically connected to the rejecting states $q_{1,0}, q_{2,0}, \ldots, q_{k,0}$. All the remaining states are accepting.

Denote $m = F(n-1) = \text{lcm}(i_1, i_2, \ldots, i_k)$. Consider the set of $m+1$ pairs of strings $\mathcal{F} = \{(\varepsilon, \varepsilon)\} \cup \{(a^i, a^{m+1-i}) \mid 1 \leq i \leq m\}$, and let us show that $\mathcal{F}$ is a fooling set for the language $L(N)^c$.

(F1) The strings ε and aa^m are not accepted by N since the initial state s is rejecting, and every computation on aa^m ends up in a rejecting state $q_{j,0}$ because each i_j divides m. Hence $\varepsilon\varepsilon$ and $a^i a^{m+1-i}$ with $1 \leq i \leq m$ are in $L(N)^c$.

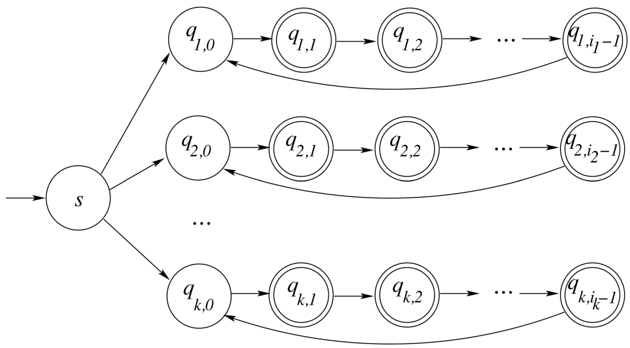

Fig. 7. A unary non-returning NFA N meeting the bound $F(n-1)+1$ for complement

(F2) If $1 \le \ell < m = \text{lcm}(i_1, i_2, \ldots, i_k)$, then some i_j does not divide ℓ. This means that the computation on aa^ℓ beginning with states s and $q_{j,0}$ ends up in an accepting state in $\{q_{j,1}, q_{j,2}, \ldots, q_{j,i_j-1}\}$. It follows that the strings $\varepsilon a^m, \varepsilon a^{m-1}, \ldots, \varepsilon a^2$ and $a^m \varepsilon$, as well as the strings $a^i a^{m+1-j} = aa^{m-(j-i)}$, where $1 \le i < j \le m$, are accepted by N, and therefore they are not in $L(N)^c$.

Thus $\mathcal{F}$ is a fooling set for the language $L(N)^c$, and the lemma follows. □

Hence we get the following result.

Theorem 5 (Complement on Unary Non-Returning NFA Languages). *Let L be a unary non-returning NFA language with* $\text{nsc}(L) = n$. *Then* $\text{nsc}(L^c)$ *is in* $2^{O(\sqrt{n \log n})}$. *The bound* $2^{\Omega(\sqrt{n \log n})}$ *can be met infinitely many times.*

Proof. Every unary n-state NFA can be simulated by a $[2n^2 + n + F(n)]$-state DFA [4, 19]. After interchanging the final and non-final states, we get a DFA for the complement with the same number of states. Since $2n^2 + n + F(n)$ is in $2^{O(\sqrt{n \log n})}$, this gives the upper bound. For the lower bound, consider the language L accepted by the n-state NFA N shown in Fig. 7.

First, we show that the language L is a non-returning NFA language. We denoted $m = F(n-1)$, thus $n < m$. Let N' be an n'-state NFA for the language $L(N)$ with $n' \le n$. Then N' must accept all strings a^i with $2 \le i \le m$ since all these strings are in $L(N)$. Assume for a contradiction that N' is not non-returning. Then, the initial state of N' is in a cycle of length ℓ with $1 \le \ell \le n' < m$. But then N' accepts the string $a^{m+1} = a^\ell \cdot a^{m+1-\ell}$ which is a contradiction since a^{m+1} is not in $L(N)$.

Now, let $k = \min\{\ell \mid F(\ell) = F(n-1)\}$. Let us show that $k \le \text{nsc}(L) \le k+1$. Recall that $m = F(n-1)$, thus $m = F(k)$. Let $F(k) = \text{lcm}(x_1, \ldots, x_r)$. Then L is accepted by a $(k+1)$-state NFA consisting of an initial state that is nondeterministically connected to r disjoint cycles of lengths $x_1, \ldots, x_r$.

Next, assume for a contradiction that L is accepted by an n'-state NFA N' with $n' < k$. Then in the Chrobak normal form of the NFA N', the number of states in cycles is at most n'. It follows that L is accepted by a DFA A, the loop of which is of length at most $F(n') < m$. Then there is an integer $\hat{t}$ such that the computation of the DFA A on the string $aa^{\hat{t}m}$ ends in the loop. However, all the strings $aa^{\hat{t}m} \cdot a^i$ with $1 \le i \le m-1$ must be accepted since they are in L. It follows that all the states in the loop of the DFA A must be final. But then the DFA A accepts a co-finite language, which is a contradiction since the language L is not co-finite. Since $F(n-1)+1$ is in $2^{\Omega(\sqrt{n \log n})}$, the theorem follows. □

6 Binary Alphabet

In this section, we study the complementation operation on binary prefix-free and suffix-free languages. We prove that the nondeterministic state complexity of complementation in this case is still exponential in $2^{\Omega(\sqrt{n \log n})}$. In the case of prefix-free binary languages, we prove that the upper bound 2^{n-1} given by Lemma 2 cannot be met. Whether or not this bound can be met by binary suffix-free languages remains open. Let us start with lower bounds.

Lemma 8. *There exists a binary prefix-free (suffix-free) n-state NFA N such that every NFA for the complement of $L(N)$ requires at least $F(n-2)+1$ states.*

Proof. Let L be the unary language accepted by an $(n-1)$-state NFA given by Lemma 7. Let $\mathcal{F} = \{(x_i, y_i) \mid i = 1, 2, \ldots, F(n-2)+1\}$ be the fooling set for L^c given in the proof of Lemma 7.

In the prefix-free case, we take an n-state NFA for the binary prefix-free language Lb. Then the set $\{(x_i, y_i\, b) \mid (x_i, y_i) \in \mathcal{F}\}$ is the fooling set for $(Lb)^c$ of size $F(n-2)+1$, and the lemma follows.

In the suffix-free case, we take an n-state NFA for the language bL. This time, the fooling set for $(bL)^c$ is $\{(b\, x_i, y_i) \mid (x_i, y_i) \in \mathcal{F}\}$. □

The next lemma provides an upper bound on the nondeterministic state complexity of complementation on binary prefix-free languages.

Lemma 9. *Let $n \geq 12$. Let L be a binary prefix-free language with* $\text{nsc}(L) = n$. *Then* $\text{nsc}(L^c) \leq 2^{n-1} - 2^{n-3} + 1$.

Proof. Let N be a minimal NFA for L. Let $\{1, 2, \ldots, n\}$ be the state set of N. Let n be the final state of N. Without loss of generality, the state n is reached from the state $n-1$ on a in N.

If there is no transition (i, a, j) with $i, j \in \{1, 2, \ldots, n-1\}$, then the automaton on states $\{1, 2, \ldots, n-1\}$ is unary. It follows that in the subset automaton of N, at most $O(F(n-1)) < 2^{n-1} - 2^{n-3}$ subsets of $\{1, 2, \ldots, n-1\}$ can be reached, and the lemma follows in this case.

Now consider a transition (i, a, j) with $i, j \in \{1, 2, \ldots, n-1\}$. Let us show that no subset of $\{1, 2, \ldots, n-1\}$ containing states i and $n-1$ may be reachable. Assume for contradiction, that a set $S \cup \{i, n-1\}$ is reached from the initial state of the subset automaton by a string u. Since N is minimal, the final state n is reached from the state j by a non-empty string v. However, the set $S \cup \{i, n-1\}$ goes to a final set $S' \cup \{j, n\}$ by a, and then to a final set $S'' \cup \{n\}$ by v. It follows that the subset automaton accepts the strings ua and uav, which is a contradiction with the prefix-freeness of the accepted language. Thus at least 2^{n-3} subsets of $\{1, 2, \ldots, n-1\}$ are unreachable. Therefore, the subset automaton has at most $2^{n-1} - 2^{n-3} + 1$ states. After exchanging the accepting and the rejecting states we get a DFA of the same size for the complement of $L(N)$, and the lemma follows. □

Now we summarize the results given by Lemma 9 and Lemma 8 in the following theorem; recall that $F(n) = \max\{\text{lcm}(x_1, \ldots, x_k) \mid x_1 + \cdots + x_k = n\}$, and that $F(n)$ is in $2^{\Theta(\sqrt{n \log n})}$.

Theorem 6 (Complement on Binary Prefix-Free Languages). *Let L be a binary prefix-free language with* $\text{nsc}(L) = n$. *Then* $\text{nsc}(L^c) \leq 2^{n-1} - 2^{n-3} + 1$. *The lower bound $F(n-2)+1$ can be met for infinitely many n.* □

References

1. Aho, A.V., Ullman, J.D., Yannakakis, M.: On notions of information transfer in VLSI circuits. In: Johnson, D.S., et al. (eds.) STOC 1983, pp. 133–139. ACM (1983)
2. Bach, E., Shallit, J.: 2.7 in Algorithmic Number Theory. Efficient Algorithms, vol. 1. MIT Press, Cambridge (1996)
3. Birget, J.C.: Partial orders on words, minimal elements of regular languages, and state complexity. Theoret. Comput. Sci. 119, 267–291 (1993), ERRATUM: Partial orders on words, minimal elements of regular languages, and state complexity (2002), `http://clam.rutgers.edu/~birget/papers.html`
4. Chrobak, M.: Finite automata and unary languages. Theoret. Comput. Sci. 47, 149–158 (1986)
5. Glaister, I., Shallit, J.: A lower bound technique for the size of nondeterministic finite automata. Inform. Process. Lett. 59, 75–77 (1996)
6. Eom, H.-S., Han, Y.-S., Jirásková, G.: State complexity of basic operations on non-returning regular languages. In: Jurgensen, H., Reis, R. (eds.) DCFS 2013. LNCS, vol. 8031, pp. 54–65. Springer, Heidelberg (2013)
7. Ellul, K.: Descriptional complexity measures of regular languages. Master's thesis. University of Waterloo (2002)
8. Han, Y.-S., Salomaa, K.: State complexity of basic operations on suffix-free regular languages. Theoret. Comput. Sci. 410, 2537–2548 (2009)
9. 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. Institute of Informatics, University of Szeged (2009)
10. Han, Y.-S., Salomaa, K., Wood, D.: Nondeterministic state complexity of basic operations for prefix-free regular languages. Fundam. Inform. 90, 93–106 (2009)
11. Han, Y.-S., Salomaa, K.: Nondeterministic state complexity for suffix-free regular languages. In: McQuillan, I., Pighizzini, G. (eds.) DCFS 2010. EPTCS, vol. 31, pp. 189–196 (2010)
12. Holzer, M., Kutrib, M.: Nondeterministic descriptional complexity of regular languages. Int. J. Found. Comput. Sci. 14, 1087–1102 (2003)
13. Hromkovič, J.: Communication complexity and parallel computing. Springer (1997)
14. Jirásková, G.: State complexity of some operations on regular languages. In: Csuhaj-Varjú, E., et al. (eds.) DCFS 2003. MTA SZTAKI, pp. 114–125. Hungarian Academy of Sciences, Budapest (2003)
15. Jirásková, G.: State complexity of some operations on binary regular languages. Theoret. Comput. Sci. 330, 287–298 (2005)
16. Maslov, A.N.: Estimates of the number of states of finite automata. Soviet Math. Doklady 11, 1373–1375 (1970)
17. Sakoda, W.J., Sipser, M.: Nondeterminism and the size of two-way finite automata. In: Proc. 10th Annual ACM Symposium on Theory of Computing, pp. 275–286 (1978)
18. Sipser, M.: Introduction to the theory of computation. PWS Publishing Company, Boston (1997)
19. To, A.W.: Unary finite automata vs. arithmetic progressions. Inform. Process. Lett. 109, 1010–1014 (2009)
20. Yu, S.: Regular languages. In: Rozenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages, vol. I, ch. 2, pp. 41–110. Springer, Heidelberg (1997)
21. Yu, S., Zhuang, Q., Salomaa, K.: The state complexity of some basic operations on regular languages. Theoret. Comput. Sci. 125, 315–328 (1994)