

# Complement on Free and Ideal Languages

Peter Mlynárčik<sup>(✉)</sup>

Mathematical Institute, Slovak Academy of Sciences,  
Grešaková 6, 040 01 Košice, Slovakia  
mlynarcik1972@gmail.com

**Abstract.** We study nondeterministic state complexity of the complement operation on the classes of prefix-free, suffix-free, factor-free and subword-free languages and on the class of ideal languages. For the cases prefix-free and suffix-free we improve the lower bound, and improve the upper bound for suffix-free languages in the binary case. In all other cases, we find tight bounds for sufficient alphabet sizes.

## 1 Introduction

The complement of a formal language  $L$  over an alphabet  $\Sigma$  is the language  $L^c = \Sigma^* \setminus L$ , where  $\Sigma^*$  is the set of all strings over an alphabet  $\Sigma$ . 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 [9] 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 [1], but later corrected this to a four-letter alphabet. Holzer and Kutrib [4] 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 [5]. In the unary case, the complexity of complementation is known to be in  $e^{\Theta(\sqrt{n \ln(n)})}$  [4, 5].

Jirásková and Mlynárčik [7] gave tight bounds in case of prefix- and suffix-free languages over a ternary alphabet and for binary languages gave the lower bound  $F(n-2) + 1$ , where  $F(n)$  is the Landau function, and  $F(n)$  is in the class  $e^{\Theta(\sqrt{n \ln(n)})}$ . The upper bound for binary alphabet was improved to  $2^{n-1} - 2^{n-3} + 1$ , but only in the prefix-free case. The suffix-free case remained open.

In this paper, we investigate the complementation operation on prefix-free and suffix-free binary languages, where we give significantly better lower bounds

---

P. Mlynárčik—Research supported by VEGA grant 2/0084/15.

$2^{\lfloor \frac{n}{2} \rfloor - 1}$  and give an improved upper bound for the suffix-free language  $2^{n-1} - 2^{n-3} + 2$ . We also deal with factor-free and subword-free languages, where we give tight bounds for proper alphabet. For the factor-free case over binary alphabets we get a result similar as to that mentioned above. In the second part of the paper we deal with complementation ideal languages, including right ideals, left ideals, two-sided ideals, and all-sided ideals. In the first three cases we give a tight bound in binary case, and in the last case the tight bound is for an exponentially-growing alphabet.

## 2 Preliminaries

Recall that 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.

To prove the minimality of nondeterministic finite automata, we use a fooling set lower-bound technique [1, 8].

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

**(F1)**  $x_i y_i \in L$ , and

**(F2)** if  $i \neq j$ , then  $x_i y_j \notin L$  or  $x_j y_i \notin L$ .

**Lemma 2** ([1, 8]). *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.  $\square$*

Although the difference between the size of fooling set and the size of minimal NFA can be large, we successfully use the fooling set technique throughout the paper [6].

Landau’s function is frequently needed, and is defined as follows:

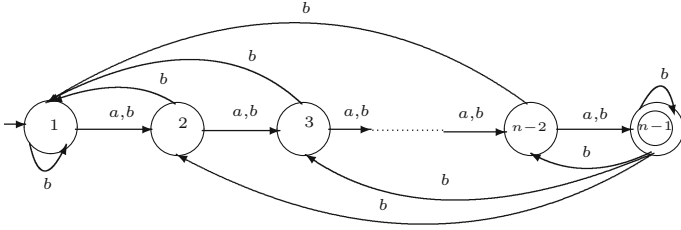
Let  $n$  be a positive integer. Then  $F(n) = \max\{\text{lcm}(x_1, x_2, \dots, x_k) \mid x_1 + x_2 + \dots + x_k = n\}$ . The function  $F(n)$  is in the class  $e^{\theta(\sqrt{n \ln(n)})}$  (Landau, 1903).

## 3 Free Languages

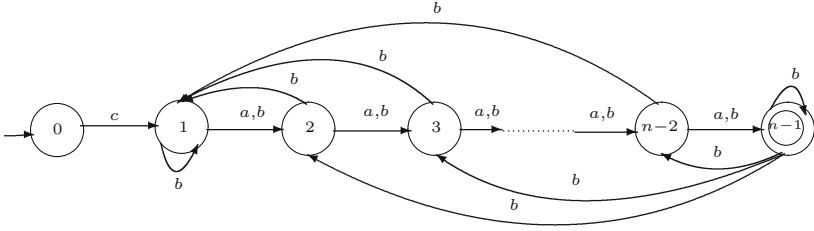
Let  $G$  be the language accepted by the NFA over  $\{a, b\}$  shown in Fig. 1 with  $n - 1$  states. Let  $L = cG$ . The language  $L$  is a suffix-free language over  $\{a, b, c\}$  recognized by an  $n$ -state NFA  $A$ , shown in Fig. 2 and  $\text{nsc}(L^c) \geq 2^{n-1}$  [7]. Now let us define a homomorphism  $h$  as follows:  $h(c) = 00$ ,  $h(a) = 10$ ,  $h(b) = 11$  (used in [2]). After applying  $h$  on the language  $L$ , we have a binary language  $K = h(L)$  over  $\{0, 1\}$ .

**Lemma 3.** *The language  $K$  is a suffix-free language.*

*Proof.* Every string in  $L$  contains exactly one symbol  $c$  at the beginning, so every string in  $K$  begins with the string  $00$  and this substring does not appear later in string. If there is a string  $w = uv$  and  $u \neq \varepsilon$ , then  $v$  does not contain  $00$  and therefore  $v \notin K$ . So  $K$  is suffix-free.  $\square$



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



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

Now let us define NFA  $A'$  for the language  $K$ . We use the description of automaton  $A$  for original language  $L$ . Let  $A = (Q, \{a, b, c\}, \delta, 0, \{n - 1\})$  (NFA shown in Fig. 2). The idea is replace every transition  $q \xrightarrow{a} q_a$  by adding a new state  $q'$  and two transitions  $q \xrightarrow{1} q' \xrightarrow{0} q_a$ , and similarly for the symbol  $b$   $q \xrightarrow{1} q' \xrightarrow{1} q_b$  and transition  $0 \xrightarrow{c} 1$  we replace by adding  $0'$  and two transitions  $0 \xrightarrow{0} 0' \xrightarrow{0} 1$ .

**Lemma 4.** *The NFA  $A'$  defined above recognizes the language  $K$ .* □

**Lemma 5.** *The NFA  $A'$  is a minimal NFA for the language  $K$ .* □

**Lemma 6.** *Let  $n \geq 3$  and  $K$  be the binary language defined above. Then  $\text{nsc}(K^c) \geq 2^{n-1}$ .*

*Proof.* As shown in [7, Lemma 5], the set  $\mathcal{F} = \{(cx_S, y_S) \mid S \subseteq \{1, 2, \dots, n - 1\}\}$  is a fooling set for  $L^c$ . Let us define  $\mathcal{F}' = \{(h(cx_S), h(y_S)) \mid S \subseteq \{1, 2, \dots, n - 1\}\}$ . Let us show that the  $\mathcal{F}'$  is a fooling set for  $K^c$ .

- (F1) For every pair  $(h(cx_S), h(y_S))$ , we have  $cx_S y_S \in L^c$ , so  $cx_S y_S \notin L$  and since homomorphism  $h$  is a bijection, we have  $h(cx_S y_S) \notin K$  so  $(h(cx_S), h(y_S)) \in K^c$ .
- (F2) Let  $(h(cx_S), h(y_S)), (h(cx_T), h(y_T))$  be two distinct pairs. Without loss of generality, let  $cx_S y_T \notin L^c$ . So  $cx_S y_T \in L$ , then  $h(cx_S y_T) \in K$ , so  $h(cx_S y_T) \notin K^c$ .

Hence  $\mathcal{F}'$  is a fooling set for  $K^c$ . Since  $|\mathcal{F}'| = 2^{n-1}$ ,  $\text{nsc}(K^c) \geq 2^{n-1}$ . □

**Proposition 7.** *Let  $L$  be a suffix-free language  $L$  over alphabet  $\Sigma$ . Then for every  $x \in \Sigma$  the language  $R = xL$  is suffix-free.*  $\square$

Above we found a binary language with an even nondeterministic state complexity, and now we want to find a binary language with an odd one. Now let us consider the language  $K_1 = 0K$ , where  $K$  is described above. By Proposition 7,  $K_1$  is a suffix-free language.

**Lemma 8.** *Let  $K$  and  $K_1$  be binary suffix-free languages mentioned above. Then  $\text{nsc}(K_1) = 2n + 1$ .*

*Proof.* Let us consider the automaton  $A'$  for the language  $K$ . Let us construct an automaton  $A''$  from  $A'$  by simply adding a new state  $0'$  and transition from  $0'$  to the original initial state  $0$  on symbol  $0$ . State  $0'$  becomes a new initial state.

Now let us consider the minimality of  $A''$ . Let  $\mathcal{F}$  be a fooling set for  $K$ . Let us construct  $\mathcal{F}'$  from  $\mathcal{F}$  as follows:  $\mathcal{F}' = \{(0u, v) \mid (u, v) \in \mathcal{F}\} \cup \{\varepsilon, 000(10)^{n-2}\}$ .

The set  $\mathcal{F}'$  is a fooling set for  $K_1$  and  $|\mathcal{F}'| = 2n + 1$ , so  $\text{nsc}(K_1) = 2n + 1$ .  $\square$

**Lemma 9.** *Let  $n \geq 3$  and  $K_1$  be the binary language defined above. Then  $\text{nsc}(K_1^c) \geq 2^{n-1}$ .*

*Proof.* Let  $\mathcal{F}$  be a fooling set for language  $K^c$  (see Lemma 6). Let us construct the set  $\mathcal{F}' = \{(0u, v) \mid (u, v) \in \mathcal{F}\}$ . Let us show that  $\mathcal{F}'$  is a fooling set for  $K_1^c$ .

(F1) If  $uv \in K^c$ , then  $uv \notin K$ , then also  $0uv \notin K_1$ , so  $0uv \in K_1^c$ .

(F2) If  $(u, v), (x, y) \in \mathcal{F}$  and without loss of generality,  $uy \notin K^c$ , so  $uy \in K$ .

Then  $0uy \in K_1$  and  $0uy \notin K_1^c$ .

Hence  $\mathcal{F}'$  is a fooling set for  $K_1^c$ . Since  $|\mathcal{F}'| = 2^{n-1}$ ,  $\text{nsc}(K_1^c) \geq 2^{n-1}$ .  $\square$

We summarize our results in the following theorem.

**Theorem 10.** *Let  $n \geq 6$ . There is a binary suffix-free language  $L$  such that  $\text{nsc}(L) = n$  and  $\text{nsc}(L^c) \geq 2^{\lfloor \frac{n}{2} \rfloor - 1}$ .*

Now we consider an upper bound. Let us recall the following result.

**Lemma 11.** *Let  $n \geq 12$ . Let  $L$  be a binary prefix-free language with  $\text{nsc}(L) = n$ . Then  $\text{nsc}(L^c) \leq 2^{n-1} - 2^{n-3} + 1$ . [7, Lemma 9]*

Notice that the proof at [7, Lemma 9] also works for NFAs with multiple initial states. We are also going to use it for suffix-free languages.

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

*Proof.* After reversing an NFA for  $L$ , we obtain an  $n$ -state NFA (possibly with multiple initial states) for a prefix-free language  $L^R$ . By Lemma 11,  $\text{nsc}((L^R)^c) \leq 2^{n-1} - 2^{n-3} + 1$ . Since  $(L^R)^c = (L^c)^R$ , we have  $\text{nsc}((L^c)^R) \leq 2^{n-1} - 2^{n-3} + 1$ . It follows that  $(L^c)^R$  is accepted by an NFA  $N$  which has at most  $2^{n-1} - 2^{n-3} + 1$  states. Now we reverse the NFA  $N$ , and get a NFA  $N^R$ , possibly with multiple initial states. By adding one more state, we get an NFA for  $L^c$  with at most  $2^{n-1} - 2^{n-3} + 2$  states and with a unique initial state.  $\square$

Similarly as in the case of suffix-free language, we can apply the same homomorphism  $h$  on the ternary prefix-free language  $L$  from [7, Lemma 3]. We only have to be careful with the proof of the prefix-free property of the language  $h(L)$ . Now every string in  $h(L)$  ends with 00. The only proper prefix of a string in  $h(L)$  which ends with 00 has an odd length. But such a string does not belong to  $h(L)$ . Therefore  $h(L)$  is prefix-free.

We can construct NFA  $A$  for  $h(L)$  with  $2n$  states similarly as in the suffix-free case. The main difference between the automaton for the case of a binary suffix-free language, and for a binary prefix-free language is the final state. Similarly as in suffix-free case we can prove that  $A$  is minimal and therefore  $\text{nsc}(h(L)) = 2n$ . Finally, we use a similar approach to find a binary prefix-free language with an odd number of states, such that we add a new state  $n'$  and the transition from original final state  $n$  to  $n'$  on symbol 0. State  $n'$  become a new final state. Such a language is still prefix-free.

Hence we get the following result for binary prefix-free languages.

When we use the result from Lemma 11 we can state the following result.

**Theorem 13 (Complement on Binary prefix-free, suffix-free languages).**

Let  $n \geq 12$ . Let  $L$  be a binary prefix-free or suffix free language with  $\text{nsc}(L) = n$ . Then  $\text{nsc}(L^c) \leq 2^{n-1} - 2^{n-3} + 2$ . The lower bound is  $2^{\lfloor \frac{n}{2} \rfloor - 1}$ .

In the paper [7, Lemma 8] we presented a binary suffix-free and prefix-free language  $L$  with  $\text{nsc}(L) \leq n$ , such that every NFA for its complement requires at least  $F(n-2)+1$  states, where  $F(n)$  is the Landau function. The function  $F(n)$  is in  $2^{\Theta(\sqrt{n \log(n)})}$ ; therefore  $\lim_{n \rightarrow \infty} F(n-2) + 1/2^{\lfloor \frac{n}{2} \rfloor - 1} = 0$ . So the lower bound in our Theorem 13 is significantly higher.

After investigation of prefix and suffix free languages we will investigate other free classes of languages: factor-free and subword-free languages. First we present a lemma which we use in our next considerations.

**Lemma 14.** Let  $L$  be a language such that  $\varepsilon \in L$ . Let  $u$  and  $v$  be strings, and let  $u \notin L$ . Let  $\mathcal{A}$  be a set of pairs of strings such that the sets  $\mathcal{A} \cup \{(\varepsilon, v)\}$  and  $\mathcal{A} \cup \{(u, v)\}$  are fooling sets for  $L$ . Then  $\text{nsc}(L) \geq |\mathcal{A}| + 2$ . □

Let  $w$  be a string. We say that a string  $v$  is a *factor* of the  $w$  iff there are strings  $x, v, y$ , such that  $w = xvy$ . Moreover, if  $xy \neq \varepsilon$ , we say that  $v$  is a *proper factor*. We say a language  $L$  is *factor-free* iff there are no two strings  $u, v$  in  $L$ , such that  $u$  is a proper factor of  $v$ .

**Theorem 15.** Let  $n \geq 3$ . Let  $L$  be a factor-free language over an alphabet  $\Sigma$  such that  $\text{nsc}(L) = n$ . Then  $\text{nsc}(L^c) \leq 2^{n-2} + 1$ , and the bound is tight if  $|\Sigma| \geq 3$ .

*Proof.* We first prove the upper bound. Let  $A$  be an  $n$ -state NFA for  $L$ . Since  $L$  is factor-free, it is suffix-free and also prefix-free. It follows that no transition goes to the initial state of  $A$ , and all the final states in the subset automaton are equivalent. Hence the subset automaton has at most  $2^{n-2} + 2$  reachable and pairwise distinguishable states. After exchanging the final and non-final states,

we get a DFA for  $L^c$  of at most  $2^{n-2} + 2$  states. In the same way as for prefix-free languages in [7, Lemma 2], we can use a nondeterminism to save one state. This gives the upper bound  $2^{n-2} + 1$ .

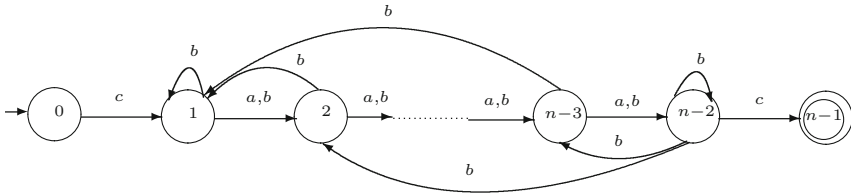
To prove tightness, consider the binary language  $G$  accepted by the  $(n - 2)$ -state NFA  $N$  shown in Fig. 1. Let  $L = c \cdot G \cdot c$ . Then  $L$  is accepted by an  $n$ -state NFA  $A$  shown in Fig. 3.

Let  $\mathcal{F} = \{(x_S, y_S) \mid S \subseteq \{1, 2, \dots, n - 2\}\}$  be a fooling set for the language  $G^c$  [5, Theorem 5]. Notice that the strings  $x_S$  and  $y_S$  have the following properties:

- (1) by  $x_S$ , the initial state goes to the set  $S$ ;
- (2) the string  $y_S$  is rejected by  $N$  from every state in  $S$  and it is accepted by  $N$  from every state in  $\{1, 2, \dots, n - 2\} \setminus S$ .

Then  $\mathcal{F}' = \{(cx_S, y_Sc) \mid S \subseteq \{1, 2, \dots, n - 2\}\}$  is a fooling set for  $L^c$ . Let  $\mathcal{A} = \{(cx_S, y_Sc) \mid S \subseteq \{1, 2, \dots, n - 2\} \text{ and } S \neq \emptyset\}$ ,  $v = y_\emptyset \cdot c$ ,  $u = ca^{n-3}c$ . Let us show that  $L^c$ ,  $\mathcal{A}$ ,  $u$  and  $v$  satisfy the conditions of Lemma 14.

First, we have  $\varepsilon \in L^c$  and  $u \notin L^c$ . Next, the string  $\varepsilon \cdot y_\emptyset \cdot c$  is in  $L^c$  since it does not begin with  $c$ . The string  $uv = ca^{n-3}c \cdot y_\emptyset c$  is in  $L^c$  since it contains three  $c$ 's. The set  $\mathcal{A}$  is a fooling set for  $L^c$  since  $\mathcal{A} \subseteq \mathcal{F}'$ . Notice that the string  $y_\emptyset c$  is accepted by  $A$  from each state in  $\{1, 2, \dots, n - 2\}$  since  $y_\emptyset$  is accepted by  $N$  from each state in  $\{1, 2, \dots, n - 2\}$  [5, Theorem 5]. Thus, if  $S$  is non-empty, then  $cx_S y_\emptyset c \notin L^c$  since by  $cx_S$  the NFA  $A$  reaches the non-empty set  $S$ , from which it accepts  $y_\emptyset c$ . It follows that the conditions in Lemma 14 are satisfied, and therefore we have  $\text{nsc}(L^c) \geq |\mathcal{A}| + 2 = 2^{n-2} + 1$ . This completes our proof. □



**Fig. 3.** An NFA of a ternary factor-free language  $L$  with  $\text{nsc}(L^c) = 2^{n-2} + 1$

It remains to find the bounds for the binary case.

Let us start with the upper bound. Let  $L$  be a binary factor-free language with  $\text{nsc}(L) = n$ , accepted by an  $n$ -state NFA  $N$ . The NFA  $N$  has to have properties as an automaton for a prefix- or suffix-free language, so there is just one final state with no outgoing transition and no transition goes to the initial state. We obtain a similar lemma as in the case of binary prefix-free languages in [7, Lemma 9].

**Lemma 16.** *There is a positive integer  $n_0$  such that for every  $n > n_0$ , if  $L$  is a binary factor-free language with  $\text{nsc}(L) = n$  then  $\text{nsc}(L^c) \leq 2^{n-2} - 2^{n-4} + 1$ .*

For the lower bound, let us consider the language  $L = cGc$ , where  $G$  is accepted by the  $(n - 2)$ -state NFA shown in Fig. 1. Then  $L$  is accepted by an  $n$ -state NFA  $A$  shown in Fig. 3. By a similar strategy as in the binary case of prefix- or suffix-free language, we apply the homomorphism  $h$  on the language  $L$ . Every string  $w$  in  $h(L)$  has the form  $001u1100$  or  $001u1000$  and the string  $u$  does not contain the string  $00$ . So in the first case, any proper factor belonging to  $h(L)$  does not exist. In the second case, every proper factor belonging to  $h(L)$  has to have the form  $001u100$  but it has an odd length, and since every string in  $h(L)$  has an even length, such a string is not in  $h(L)$ . So  $h(L)$  is factor-free. We get an NFA  $A$  for  $h(L)$  in a similar way as in the suffix-free or prefix-free cases. The NFA  $A$  is minimal and has  $2n$  states, so  $\text{nsc}(h(L)) = 2n$ .

We deal with odd values of  $n$  similarly as before. Thus we get the following result.

**Lemma 17.** *Let  $n \geq 8$ . There is a binary factor-free language  $L$  such that  $\text{nsc}(L) = n$  and  $\text{nsc}(L^c) \geq \Omega(2^{\frac{n}{2}})$ .*

We summarize our results about binary factor-free languages in the following theorem.

**Theorem 18.** *There is a positive integer  $n_0$  such that for every  $n > n_0$ , if  $L$  is a binary factor-free language with  $\text{nsc}(L) = n$  then  $\text{nsc}(L^c) \leq 2^{n-2} - 2^{n-4} + 1$ . The lower bound is  $\Omega(2^{\frac{n}{2}})$ .*

Let  $w$  be a string such that  $w = u_0v_1u_1v_2u_2 \cdots v_mu_m$ , where every  $u_i$  and  $u_j$  are strings in  $\Sigma^*$ . We say that the string  $v = v_1v_2 \cdots v_m$  is a *subword* of the  $w$ . Moreover if  $v \neq w$ , we say that  $v$  is a *proper subword*.

For example let  $w = abacb$ . Strings  $abac, bbb, bc$  are subwords of  $w$ , but the string  $aca$  is not a subword of  $w$ .

Let  $L$  be a language. We say  $L$  is *subword-free* iff there are no two strings  $u, v$  in  $L$  such that  $u$  is a proper subword of  $v$ .

**Proposition 19.** *Let  $L$  be a language. If  $L$  is subword-free, then  $L$  is finite.*

**Theorem 20.** *Let  $n \geq 4$ . Let  $L$  be a subword-free language over an alphabet  $\Sigma$  such that  $\text{nsc}(L) = n$ . Then  $\text{nsc}(L) \leq 2^{n-2} + 1$ , and the bound is tight if  $|\Sigma| \geq 2^{n-2}$ .*

*Proof.* The upper bound is the same as for factor-free languages. To prove tightness, let  $\Sigma = \{a_S \mid S \subseteq \{1, 2, \dots, n - 2\}\}$  be an alphabet with  $2^{n-2}$  symbols.

Consider the language  $L$  accepted by the NFA  $A = (Q, \Sigma, \delta, 0, \{n - 1\})$ , where  $Q = \{0, 1, \dots, n - 1\}$ , and the transition function  $\delta$  is defined as follows: for each symbol  $a_S$  in  $\Sigma$ ,  $\delta(0, a_S) = S$ ;  $\delta(i, a_S) = \emptyset$  if  $1 \leq i \leq n - 2$  and  $i \in S$ ;  $\delta(i, a_S) = \{n - 1\}$  if  $1 \leq i \leq n - 2$  and  $i \notin S$ ; and  $\delta(n - 1, a_S) = \emptyset$ .

Notice that each string in  $L$  is of length 2, so  $L$  is subword-free. Consider the set of pairs  $\mathcal{F} = \{(a_S, a_S) \mid S \subseteq \{1, 2, \dots, n - 2\}\}$ . Let us show that the set  $\mathcal{F}$  is a fooling set for  $L^c$ .

- (F1) For each  $S$ , the string  $a_S a_S$  is in  $L^c$ , since  $A$  goes to  $S$  by  $a_S$  and  $a_S$  is rejected by  $A$  from each state in  $S$ .
- (F2) Let  $S \neq T$ . Then, without loss of generality, there is a state  $q$  in  $\{1, 2, \dots, n-2\}$  such that  $q \in S$  and  $q \notin T$ . Then  $a_S a_T$  is not in  $L^c$  since  $A$  goes to the state  $q$  by  $a_S$ , and then to the accepting state  $n-1$  by  $a_T$ .

Hence  $\mathcal{F}$  is a fooling set for  $L^c$ .

Let  $\mathcal{A} = \{(a_S, a_S) \mid \emptyset \neq S \subseteq \{1, 2, \dots, n-2\}\}$ ,  $u = a_{\{1\}} a_{\{2\}}$ ,  $v = a_\emptyset$ . Let us show that  $L^c$ ,  $\mathcal{A}$ ,  $u$ , and  $v$  satisfy the condition in Lemma 14. First, we have  $\varepsilon \in L^c$  and  $u \notin L^c$ . Next, we have  $\varepsilon \cdot v \in L^c$  since it is a one-symbol string, and  $uv \in L^c$  since it is of length 3. Finally, notice that  $a_\emptyset$  is accepted from each state in  $\{1, 2, \dots, n-2\}$ . It follows that if  $S \neq \emptyset$ , then  $a_S a_\emptyset$  is accepted by  $A$ , so it is not in  $L^c$ . Hence  $\mathcal{A} \cup \{(\varepsilon, v)\}$  and  $\mathcal{A} \cup \{(u, v)\}$  are fooling sets for  $L^c$ . By Lemma 14, we have  $\text{nsc}(L^c) \geq 2^{n-2} + 1$ .  $\square$

Let us now consider the case for unary alphabets. An arbitrary free language  $L$  can contain only one string. We have  $L = \{a^n\}$  for some fixed natural number  $n \geq 0$ . The complement of  $L$  consists of every string with length different from  $n$ . We can extend the theorem in [7, Theorem 4] by a more general theorem about every free language.

**Theorem 21.** *Let  $L$  be a unary prefix-free or suffix-free or factor-free or subword-free language with  $\text{nsc}(L) = n$ . Then  $\text{nsc}(L^c) = \Theta(\sqrt{n})$ .*

*Proof.* The proof is the same as in [7, Lemma 6].  $\square$

## 4 Complement on Ideal Languages

**Definition 22.** *Let  $L$  be a language over an alphabet  $\Sigma$ . Then we have four classes of ideals.*

- (1) *The language  $L$  is a right ideal iff  $L = L\Sigma^*$ .*
- (2) *The language  $L$  is a left ideal iff  $L = \Sigma^*L$ .*
- (3) *The language  $L$  is two-sided ideal iff  $L = \Sigma^*L\Sigma^*$ .*
- (4) *The language  $L$  is all-sided ideal iff  $L = L\sqcup\Sigma^*$ , where operation  $\sqcup$  is shuffle operation.*

The next proposition describes the form of a minimal NFA for some right ideal language.

**Proposition 23.** *Let  $L$  be a language over  $\Sigma$  and let  $A$  be a minimal NFA such that  $L(A) = L$ . The language  $L$  is a right ideal if and only if  $A$  contains just one final state with a loop on every letter of alphabet  $\Sigma$ .*

**Theorem 24.** *Let  $n \geq 3$ . Let  $L$  be a right ideal over an alphabet  $\Sigma$  such that  $\text{nsc}(L) = n$ . Then  $\text{nsc}(L) \leq 2^{n-1}$ , and the bound is tight if  $|\Sigma| \geq 2$ .*



*Proof.* Let  $A = (Q, \Sigma, \delta, s, F)$  be a minimal  $n$ -state NFA for a right ideal  $L$ . Then by Proposition 23 the NFA  $A$  has a unique final state  $f$  which goes to itself on every input symbol; that is, we have  $\delta(f, a) = \{f\}$  for each  $a$  in  $\Sigma$ . It follows that in the subset automaton of the NFA  $A$ , all final states are equivalent since they accept all the strings in  $\Sigma^*$ . Hence the subset automaton has at most  $2^{n-1} + 1$  reachable and pairwise distinguishable states. By interchanging the final and non-final states, we get a DFA  $B$  for  $L^c$ . The DFA  $B$  has a dead state. After removing the dead state, we get an NFA  $N$  for  $L^c$  of at most  $2^{n-1}$  states.

To prove tightness, let  $L = G \cdot b \cdot (a + b)^*$ , where  $G$  is the language accepted by the binary  $(n - 1)$ -state NFA  $N$  shown in Fig. 1. Then  $L$  is a right-ideal. The NFA  $N$  is minimal because  $\mathcal{F} = \{(a^i, a^{n-2-i}b) \mid 0 \leq i \leq n - 2\} \cup \{(a^{n-2}b, \varepsilon)\}$  is a fooling set for  $L$ .

Let  $\mathcal{F} = \{(u_S, v_S) \mid S \subseteq \{1, 2, \dots, n - 1\}\}$  be a fooling set for  $G^c$  as described in [5, Theorem 5]. We prove that the set  $\mathcal{F}' = \{(u_S, v_S \cdot b) \mid S \subseteq \{1, 2, \dots, n - 1\}\}$  is a fooling set for  $L^c$ .

- (F1) For each  $S$ , the string  $u_S v_S$  is in  $G^c$ , so it is not accepted by  $N$ . It follows that the string  $u_S v_S b$  is not accepted by  $A$ . Thus  $u_S v_S b$  is in  $L^c$ .
- (F2) Let  $S \neq T$ . Then  $u_S v_T \notin G^c$  or  $u_T v_S \notin G^c$ . In the former case, the string  $u_S v_T$  is accepted by the NFA  $N$ , and therefore the string  $u_S v_T b$  is accepted by  $A$ . Hence  $u_S v_T b \notin L^c$ . The latter case is symmetric.

Hence  $\mathcal{F}'$  is a fooling set for  $L^c$ , which means that  $\text{nsc}(L) = 2^{n-1}$ .  $\square$

The next proposition describes the form of a minimal NFA for some left ideal languages.

**Proposition 25.** *Let  $L$  be a language over  $\Sigma$  and let  $A$  be a minimal NFA such that  $L(A) = L$ . The language  $L$  is a left ideal if and only if there is a minimal NFA  $A$  in which the initial state has a loop on every input.*

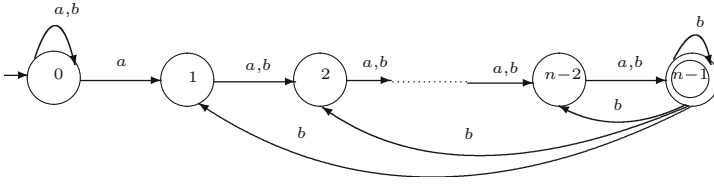
**Theorem 26.** *Let  $n \geq 3$ . Let  $L$  be a left ideal over an alphabet  $\Sigma$  such that  $\text{nsc}(L) = n$ . Then  $\text{nsc}(L) \leq 2^{n-1}$ , and the bound is tight if  $|\Sigma| \geq 2$ .*

*Proof.* Let  $A = (Q, \Sigma, \delta, s, F)$  be a minimal  $n$ -state NFA for a left ideal  $L$ . By Proposition 25 we can add a loop on the initial state  $s$  on every input symbol, we get an NFA  $N$  which is equivalent to  $A$ . Since the initial state  $s$  of  $N$  goes to itself on every input symbol, each reachable subset of the subset automaton of  $N$  contains the initial state  $s$ , so the number of all reachable subsets is at most  $2^{n-1}$ .

To prove tightness, let the language  $L$  be accepted by NFA  $A$  in Fig. 4. Then  $L$  is a binary left ideal by Proposition 25. The NFA  $A$  is minimal because  $\mathcal{F} = \{(a^i, a^{n-1-i}) \mid 0 \leq i \leq n - 1\}$  is a fooling set for  $L$ .

We are going to consider  $L^c$ . Let  $\mathcal{F} = \{(u_S, v_S) \mid S \subseteq \{1, 2, \dots, n - 1\}\}$ , where string  $u_S$  is such that the state 1 goes to the set  $S$  after reading  $u_S$  in NFA  $A$  and the string  $v_S$  is such that it is rejected by the NFA from every state  $p \in S$  and it is accepted by the NFA from every state  $p \notin S$  for any subset  $S$ .

Now, we prove that the set  $\mathcal{F}' = \{(a \cdot u_S, v_S) \mid S \subseteq \{1, 2, \dots, n - 1\}\}$  is a fooling set for  $L^c$ .



**Fig. 4.** An NFA of a binary left ideal language  $L$  with  $\text{nsc}(L^c) = 2^{n-1}$

- (F1) For each  $S$ , the string  $u_S v_S$  is not accepted from state 1, so it follows that the string  $au_S v_S$  is not accepted by  $A$ . Thus  $au_S v_S$  is in  $L^c$ .
- (F2) Let  $S \neq T$ . Then  $u_S v_T \notin L^c$  or  $u_T v_S \notin L^c$ . Let  $u_S v_T$  be accepted by the NFA  $A$ , and therefore the string  $au_S v_T$  is accepted by  $A$ . Hence  $au_S v_T \notin L^c$ . The latter case is symmetric.

Hence  $\mathcal{F}'$  is a fooling set for  $L^c$ , which means that  $\text{nsc}(L) = 2^{n-1}$ . □

**Proposition 27.** *Let  $L$  be a language over  $\Sigma$  and let  $A$  be a minimal NFA such that  $L(A) = L$ . The language  $L$  is a two-sided ideal if and only if there is a minimal NFA  $A$  with an initial state with a loop on every input and just one final state with a loop on every input.*

*Proof.* A language  $L$  is two-sided ideal if and only if it is left ideal and right ideal; therefore, the proposition follows from Propositions 23, 25. □

**Theorem 28.** *Let  $n \geq 3$ . Let  $L$  be a two-sided ideal over an alphabet  $\Sigma$  such that  $\text{nsc}(L) = n$ . Then  $\text{nsc}(L) \leq 2^{n-2}$ , and the bound is tight if  $|\Sigma| \geq 2$ .*

**Proposition 29.** *Let  $L$  be a language over  $\Sigma$ . The language  $L$  is an all-sided ideal if and only if there is a minimal NFA  $A$  with just one final state and with a loop in every state on every letter of an alphabet  $\Sigma$ , such that  $L(A) = L$ .*

We can notice that it is not necessary to have a loop for every state on every input symbol. For example a minimal NFA for the binary language  $L$  with strings of length at least 3 does not need to have loops on every states except the final one.

**Theorem 30.** *Let  $n \geq 3$ . Let  $L$  be an all-sided ideal over an alphabet  $\Sigma$  such that  $\text{nsc}(L) = n$ . Then  $\text{nsc}(L^c) \leq 2^{n-2}$ , and the bound is tight if  $|\Sigma| \geq 2^{n-2}$ .*

*Proof.* The upper bound is the same as for two-sided ideals. To prove tightness, let  $\Sigma = \{a_S \mid S \subseteq \{1, 2, \dots, n-2\}\}$  be an alphabet with  $2^{n-2}$  symbols. Consider the language  $L$  accepted by the NFA  $A = (\{0, 1, \dots, n-1\}, \Sigma, \delta, 0, \{n-1\})$  where for each symbol  $a_S$ , we have  $\delta(0, a_S) = \{0\} \cup S$ ;  $\delta(i, a_S) = \{i\}$  if  $i \in S$ ;  $\delta(i, a_S) = \{i, n-1\}$  if  $i \in \{1, 2, \dots, n-2\} \setminus S$ ;  $\delta(n-1, a_S) = \{n-1\}$ .

Since in each state of  $A$ , we have a loop on every input symbol, the language  $L$  is an all-sided ideal by Proposition 29.

Let  $\mathcal{F} = \{(a_S, a_S) \mid S \subseteq \{1, 2, \dots, n-2\}\}$ . Let us show that  $\mathcal{F}$  is a fooling set for  $L^c$ .

- (F1) For each  $S$ , the NFA  $A$  reaches the set  $\{0\} \cup S$  by  $a_S$ . By the next  $a_S$ , the NFA  $A$  remains in the set  $\{0\} \cup S$ , and rejects. Thus  $a_S a_S \in L^c$ .
- (F2) Let  $S$  and  $T$  be two subsets of  $\{1, 2, \dots, n - 2\}$  with  $S \neq T$ . Without loss of generality, there is a state  $i$  with  $i \in S$  and  $i \notin T$ . By  $a_S$ , the NFA  $A$  goes to  $\{0\} \cup S$ . Since  $i \in S$ , the NFA  $A$  goes to  $i$  by  $a_S$ . Then it goes to the state  $n - 1$  by  $a_T$  since  $i \notin T$ . Hence  $A$  accepts  $a_S a_T$ , and therefore  $a_S a_T \notin L^c$ .

Thus  $\mathcal{F}$  is a fooling set for  $L^c$ . It follows that  $\text{nsc}(L^c) \geq 2^{n-2}$ . □

Let us consider a unary alphabet. Every type of ideal language has the form  $L = \{a^k \mid k \geq n\}$ , where  $n$  is some fixed natural number. Thus, every minimal NFA  $A$  for every type of an ideal language  $L$  has a tail of  $n - 1$  states ending by final state with a loop (see the example in Fig. 5). Such an automaton  $A$  is a DFA, so after exchanging of finality and nonfinality of states we get the DFA  $A'$  with every state final except one, which is the dead state. After leaving the dead state we get the NFA  $B$  with  $n - 1$  states accepting a complement  $L^c$ .

These considerations can be summarized in the following theorem.

**Theorem 31.** *Let  $L$  be ideal over an unary alphabet, such that  $\text{nsc}(L) = n$ . Then  $\text{nsc}(L^c) = n - 1$ .*

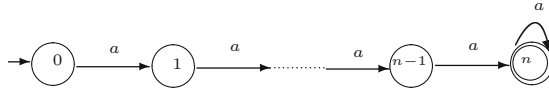


Fig. 5. Minimal NFA for language  $a^k, k \geq n$

## 5 Conclusions

Let us summarize our results. Let  $L$  be a language such that  $\text{nsc}(L) = n$ .

Firstly, let us review the case for alphabets of size 3 or more. For the suffix-free and prefix-free cases the results come from [7]. The bounds  $2^{n-1}$  are tight in both cases. For the case of factor-free, the bounds  $2^{n-2} + 1$  are tight. For the case of subword-free the upper bound is  $2^{n-2} + 1$  and it is tight when  $|\Sigma| \geq 2^{n-2}$ .

Secondly, let us review the case for binary free languages. For suffix-free languages, the lower bound is  $2^{\lfloor \frac{n}{2} \rfloor - 1}$  and upper bound is  $2^{n-1} - 2^{n-3} + 2$ , for prefix-free languages the lower bound is the same as in case of suffix-free and the upper bound [7, Lemma 9] is  $2^{n-1} - 2^{n-3} + 1$ , for factor-free the lower bound is  $\Omega(2^{\frac{n}{2}})$  and the upper bound is  $2^{n-2} - 2^{n-4} + 1$ .

For right and left ideals the bounds  $2^{n-1}$  are tight. For two-sided ideals the bounds  $2^{n-2}$  are tight. For all-sided languages the upper bound is  $2^{n-2}$  and it is tight when  $|\Sigma| \geq 2^{n-2}$ .

Finally, we will discuss the case for unary alphabets. In this case, the situation is the same for every class is : the lower bound is  $\Theta(\sqrt{n})$  and the upper bound is  $\Theta(\sqrt{n})$ .

For ideals, the situation is the same:  $\text{nsc}(L^c) = n - 1$ .

The possibility of improving the bounds for binary cases for prefix-, suffix- and factor-free languages remains open. Also in the case for subword-free languages it remains to solve the binary case. Also the possibility of finding non-exponential alphabets for witness languages for the lower bound in the case of subword-free languages remains open. The possibility of finding non-exponential alphabet for witness language for lower bound in case of all-sided ideal languages remains open.

**Acknowledgements.** I would like to thank to Galina Jirásková for many helpful discussions which led to writing this article.

## References

1. 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>
2. 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)
3. Chrobak, M.: Finite automata and unary languages. *Theoret. Comput. Sci.* **47**, 149–158 (1986)
4. Holzer, M., Kutrib, M.: Nondeterministic descriptonal complexity of regular languages. *Int. J. Found. Comput. Sci.* **14**, 1087–1102 (2003)
5. Jirásková, G.: State complexity of some operations on binary regular languages. *Theoret. Comput. Sci.* **330**, 287–298 (2005)
6. Jirásková, G.: Note on minimal automata and uniform communication protocols. In: *Grammars and Automata for String Processing: From Mathematics and Computer Science to Biology, and Back: Essays in Honour of Gheorghe Paun*, pp. 163–170. Taylor and Francis (2003)
7. Jirásková, G., Mlynářčík, P.: Complement on prefix-free, suffix-free, and non-returning NFA languages. In: Jürgensen, H., Karhumäki, J., Okhotin, A. (eds.) *DCFS 2014. LNCS*, vol. 8614, pp. 222–233. Springer, Heidelberg (2014)
8. Glaister, I., Shallit, J.: A lower bound technique for the size of nondeterministic finite automata. *Inform. Process. Lett.* **59**, 75–77 (1996)
9. Sakoda, W.J., Sipser, M.: Nondeterminism and the size of two-way finite automata. In: *Proceedings of the 10th Annual ACM Symposium on Theory of Computing*, pp. 275–286 (1978)
10. Sipser, M.: *Introduction to the Theory of Computation*. PWS Publishing Company, Boston (1997)