# Basic Operations on Binary Suffix-Free Languages*

Roland Cmorik[1] and Galina Jirásková[2]

[1] Institute of Computer Science, P.J. Šafárik University, Jesenná 5, 041 54 Košice, Slovakia
roland.cmorik@gmail.com

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

**Abstract.** We give a characterization of nondeterministic automata accepting suffix-free languages, and a sufficient condition on deterministic automata to accept suffix-free languages. Then we investigate the state complexity of basic operations on binary suffix-free regular languages. In particular, we show that the upper bounds on the state complexity of all the boolean operations as well as of Kleene star are tight in the binary case. On the other hand, we prove that the bound for reversal cannot be met by binary languages. This solves several open questions stated by Han and Salomaa (Theoret. Comput. Sci. 410, 2537-2548, 2009).

## 1 Introduction

A language is suffix-free if it does not contain two strings, one of which is a proper suffix of the other. Motivating by suffix-freeness property of some codes used in information processing and data compression, Han and Salomaa [8] examined state complexity of basic operations on suffix-free regular languages. This is a part of research devoted to investigation of the state complexity of regular operations in various subclasses of the class of regular languages [1,3,4,5,7,9,10].

Here we continue this research, and study the class of suffix-free languages in more detail. We first give a characterization of nondeterministic finite automata recognizing suffix-free languages. Using this characterization we state a sufficient condition on a deterministic finite automaton to accept a suffix-free language. This allows us to avoid proofs of suffix-freeness of languages throughout the paper. Then we study the state complexity of operations in the class of binary suffix-free languages. In particular, we show that the bounds for all the boolean operations as well as for Kleene star are tight in the binary case. On the other hand, the bound for reversal, that is tight in the ternary case [8], cannot be met by binary languages. We provide lower and upper bounds on the state complexity of reversal of binary suffix-free languages. In the case of concatenation, where witness languages in [8] are defined over a four-letter alphabet, we give ternary worst-case languages. We conclude the paper with several open problems.

* Supported by VEGA grants 1/0035/09 and 2/0183/11, and grant APVV-0035-10.

Z. Kotásek et al. (Eds.): MEMICS 2011, LNCS 7119, pp. 94–102, 2012.

## 2 Suffix-Free Languages and Suffix-Free Automata

We assume that the reader is familiar with basic notions of formal languages and automata theory, and for all unexplained notions, we refer to [13,14]. State complexity of a regular language $L$, $\mathrm{sc}(L)$, is the smallest number of states in any *complete* deterministic finite automaton (dfa) recognizing language $L$. Nondeterministic finite automata (nfa's) throughout the paper are $\varepsilon$-free.

If $w = uv$, then $v$ is a *suffix* of $w$, and if, moreover, $v \neq w$, then $v$ is a *proper suffix* of $w$. A language $L$ is *suffix-free* if for every string $w$ in $L$, no proper suffix of $w$ is in $L$. An automaton is *suffix-free* if it accepts a suffix-free language.

If an automaton accepts a non-empty suffix-free language, then it is *non-returning*, that is, no transition goes to the initial state [8]. A suffix-free minimal dfa must have a *dead state*, that is, a rejecting state that goes to itself on every symbol [8]. Our first theorem provides a characterization of suffix-free nfa's. Then, a lemma providing a sufficient condition for a dfa to be suffix-free follows. We use the lemma several times to prove the suffix-freeness of automata.

**Theorem 1 (Characterization of Suffix-Free NFA's).** *Consider a non-returning nfa without unreachable states and with the initial state $s$. Let $L_q$ be the set of strings accepted by the nfa from state $q$. The nfa accepts a suffix-free language if and only if for each state $q$ with $q \neq s$, the language $L_s \cap L_q$ is empty.*

*Proof.* Let a non-returning nfa accept a suffix-free language. Assume for a contradiction that there is a state $q$ with $q \neq s$ such that $L_s \cap L_q \neq \emptyset$. Then there exists a string $w$ that is accepted by the nfa from both $s$ and $q$. Since state $q$ is reachable, the initial state $s$ goes to state $q$ by a non-empty string $u$. Then the nfa accepts both strings $w$ and $uw$, which is a contradiction. The converse can be proved by contradiction in a similar way. □

**Lemma 1.** *Consider a non-returning dfa without unreachable states and with the sole final state. If there do not exist two distinct states that go to the same useful state by the same symbol of the input alphabet, then the dfa accepts a suffix-free language.*

*Proof.* By Theorem 1, if the dfa is not suffix-free, then there exists a string $w$ accepted from the initial state and also from some other state. The two accepting paths end in the sole final state. So these paths must meet in some useful state $q$. The two predecessors of $q$ on the two paths go to $q$ by the same symbol. □

## 3 Basic Operations on Binary Suffix-Free Languages

Han and Salomaa investigated the upper bounds for complexity of Kleene star, reversal and concatenation in [8]. They also presented witness languages for these bounds, however, the problem of tightness for small alphabets remained open. Here we investigate the complexity of mentioned operations on small alphabets.

First, we study the complexity of difference ($K \setminus L$) and symmetric difference ($K \oplus L$) that were not considered in [8]. Here we use the same witness binary languages as Olejár has used in [9] for union and intersection.

**Theorem 2 (Boolean Operations).** *Let $K$ and $L$ be suffix-free languages over an alphabet $\Sigma$ with* $\mathrm{sc}(K) = m$ *and* $\mathrm{sc}(L) = n$*, where* $m, n \geq 4$*. Then*

1. $\mathrm{sc}(K \cap L) \leq mn - 2(m + n - 3)$*;*
2. $\mathrm{sc}(K \cup L), \mathrm{sc}(K \oplus L) \leq mn - (m + n - 2)$*;*
3. $\mathrm{sc}(K \setminus L) \leq mn - (m + 2n - 4)$*.*

*All the bounds are tight if* $|\Sigma| \geq 2$*.*

*Proof.* The cases of intersection and union hold according to [9]. Let $Q_K$, $Q_L$, $F_K$, $F_L$, $s_K$, $s_L$ and $d_K$, $d_L$ denote the sets of states, final states, the initial states, and the dead states of suffix-free dfa's for $K$ and $L$, respectively. Consider the cross-product automaton for difference and symmetric difference, respectively. They differ only in final states. The set of final states is $(F_K \times Q_L) \setminus (F_K \times F_L)$ for difference and $(F_K \times Q_L \cup Q_K \times F_L) \setminus (F_K \times F_L)$ for symmetric difference. No pair $(s_K, q)$ with $q \neq s_L$ and $(p, s_L)$ with $p \neq s_K$ is reachable in the cross-product automaton since the two dfa's are non-returning. So we can remove these $m + n - 2$ unreachable states. Moreover, in the case of difference, there is no string accepted from a state, the first component of which is $d_K$. We can replace all such $n - 2$ states with one dead state. Therefore, the minimal dfa for difference has at most $mn - (m + n - 2) - (n - 2)$ states, and for symmetric difference at most $mn - (m + n - 2)$ states.

For tightness, consider the languages $K$ and $L$ accepted by dfa's $A$ and $B$ from [9], shown in Fig. 1, where dead states $m$ and $n$, as well as all the transitions to dead states, are omitted. By Lemma 1, both languages are suffix-free.

Consider the cross-product automaton for the language $K - L$, where the set of final states is $\{(m - 1, j) \mid j \neq n - 1\}$. In the proofs of Lemma 6 and Lemma 7 in [9], it is shown that states $(i, j)$ for $i = 2, \ldots, m$ and $j = 2, \ldots, n$, and the initial state $(1, 1)$ are reachable. We show that these states are pairwise distinguishable. State $(m, n)$ is the only dead state. State $(1, 1)$ is distinguished from any other state by a string starting with $b$. Consider two distinct states $(i, j)$ and $(k, \ell)$, where $2 \leq i, k \leq m - 1$, and $2 \leq j, \ell \leq n$. If $i < k$, then the string $a^n b^{m-1-k}$ is accepted from $(k, \ell)$ and rejected from $(i, j)$. If $i = k$, then we can move the two states into two distinct states $(m-1, j')$ and $(m-1, \ell')$ in row $m - 1$ by a word in $b^*$. If $j' < \ell'$, then the string $a^{n-1-j'}$ is rejected from state $(m - 1, j')$ and accepted from state $(m - 1, \ell')$. This proves distinguishability of all the $mn - (m + 2n - 4)$ states.

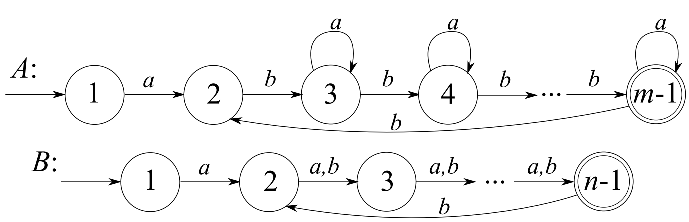


**Fig. 1.** Binary suffix-free dfa's meeting the upper bounds for Boolean operations

In the case of symmetric difference $K \oplus L$, the set of final states of the cross-product automaton is $\{(i,j) \mid i = m-1 \text{ or } j = n-1\} - \{(m-1, n-1)\}$. The proof of reachability is the same as above. State $(m, n)$ is the only dead state, and state $(1, 1)$ is distinguished from any other state by a string starting with $b$. State $(m-1, n-1)$ is distinguished from any other rejecting state by string $a^m$. Consider two distinct rejecting states $(i, j)$ and $(k, \ell)$, both different from $(m-1, n-1)$. If $i = k$ and $j < \ell$, then they can be distinguished by $a^{n-1-j}$. If $i < k$, then string $b^{m-1-i}a^m$ distinguishes them. Now consider two distinct accepting states. States $(m-1, n-2)$ and $(m-2, n-1)$ can be distinguished by $aa$. Every other pair of accepting states can be distinguished by $b$ since either one state of the pair goes to an accepting state and the second one to a rejecting state, or both go to different rejecting and, as shown above, distinguishable states. This concludes our proof. $\square$

The next theorem shows that the upper bound $2^{n-2} + 1$ for Kleene star, shown to be tight for a four-letter alphabet [8], is tight even in the binary case.

**Theorem 3 (Star).** *Let $L$ be a suffix-free language over an alphabet $\Sigma$ with* $\mathrm{sc}(L) = n$, *where* $n \geq 6$. *Then* $\mathrm{sc}(L^*) \leq 2^{n-2} + 1$. *The bound is tight if* $|\Sigma| \geq 2$.

*Proof.* The upper bound is from [8]. For tightness, consider the binary dfa $A$ depicted in Fig. 2, where $n \geq 6$. By Lemma 1, automaton $A$ is suffix-free.

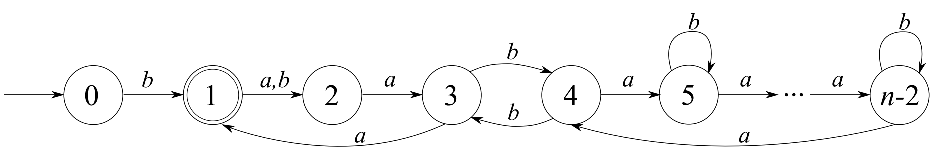


**Fig. 2.** Binary suffix-free dfa meeting the bound $2^{n-2} + 1$ for star

According to [8], we can obtain an nfa for $L(A)^*$ from automaton A by adding a new transition from state 1 to itself by $b$, and making the initial state final. Furthermore we can omit the dead state. Let us denote the obtained $(n-1)$-state nfa for $L(A)^*$ by $A'$. If we omit the initial state of nfa $A'$, and consider state 1 as the initial state, then we get an $(n-2)$-state nfa which is isomorphic to the reverse of the $(n-2)$-state Šebej's automaton [12] meeting the upper bound $2^{n-2}$ for reversal. This means that in the subset automaton corresponding to nfa $A'$, all the subsets of $\{1, 2, \ldots, n-2\}$ are reachable and pairwise distinguishable. The initial state of the subset automaton is state $\{0\}$, which is final. The string $a^3$ distinguishes state $\{0\}$ from any other final state. $\square$

Now we investigate the state complexity of concatenation of two suffix-free languages. The upper bound is $(m-1)2^{n-2} + 1$ by [8], where its tightness for a four-letter alphabet is also proved. We start with the ternary case.

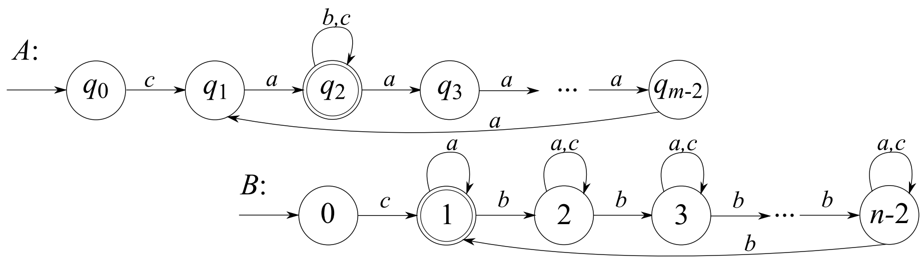


**Fig. 3.** Suffix-free dfa's meeting the bound $(m-1)2^{n-2}+1$ for concatenation

**Theorem 4 (Concatenation: Ternary Case).** *Let $K$ and $L$ be suffix-free languages over an alphabet $\Sigma$ with $\mathrm{sc}(K) = m$ and $\mathrm{sc}(L) = n$, where $m \geq 4$, $n \geq 3$. Then $\mathrm{sc}(KL) \leq (m-1)2^{n-2}+1$, and the bound is tight if $|\Sigma| \geq 3$.*

*Proof.* The upper bound is from [8]. For tightness, consider ternary regular languages $K$ and $L$ accepted by the dfa's $A$ and $B$ shown in Fig. 3; to keep the figure transparent, we omit the dead states $q_{m-1}$ and $n-1$, and all the transitions to the dead states. By Lemma 1, languages $K$ and $L$ are suffix-free.

Construct an nfa for language $KL$ from dfa's $A$ and $B$ by adding the transition on $c$ from state $q_2$ to state 1 and by declaring $q_2$ as a rejecting state.

The initial state of the corresponding subset automaton is $\{q_0\}$. We first show that for every subset $X$ of $\{1, 2, \ldots, n-2\}$, state $\{q_2\} \cup X$ is reachable. The proof is by induction on $|X|$. The basis, $|X| = 0$, holds since $\{q_2\}$ is reached from $\{q_0\}$ by $ca$. Assume that for every subset $Y$ of $\{1, 2, \ldots, n-2\}$ of size $k-1$ state $\{q_2\} \cup Y$ is reachable. Let

$$X = \{j_1, j_2, \ldots, j_k\} \text{ with } j_1 < j_2 < \cdots < j_k$$

be a subset of $\{1, 2, \ldots, n-2\}$ of size $k$. Let

$$Y = \{j_2 - j_1 + 1, \ldots, j_k - j_1 + 1\}.$$

Then state $\{q_2\} \cup Y$ is reachable by the induction hypothesis. Next, state $\{q_2\} \cup Y$ goes to $\{q_2\} \cup X$ by $cb^{j_1-1}$. Now if $i \in \{1, \ldots, m-2\}$, then state $\{q_i\} \cup X$ is reached from state $\{q_2\} \cup X$ by string $a^{m-4+i}$. State $\{q_{m-1}\} \cup X$ is reached from state $\{q_2\} \cup X$ by $b^{n-3}ab$. This proves the reachability of $(m-1)2^{n-2}+1$ states.

It remains to show that these states are pairwise distinguishable. Since string $cac$ is accepted by the nfa only from state $q_0$, the initial state $\{q_0\}$ of the subset automaton is distinguishable from any other state. States $\{q_i\} \cup X$ and $\{q_j\} \cup Y$ with $i < j$ are distinguished by $a^{m-i}c$. Finally, two states $\{q_i\} \cup X$ and $\{q_i\} \cup Y$ with $X \neq Y$ differ in a state $j$ in $\{1, 2, \ldots, n-2\}$, and so the string $b^{n-j-1}$ distinguishes the two states. □

Next we investigate the binary case. We present an $m$-state dfa and an $n$-state dfa such that the state complexity of $L(A)L(B)$ is $(m-1)2^{n-2}$ providing that $m-2$ and $n-2$ are relatively prime numbers.

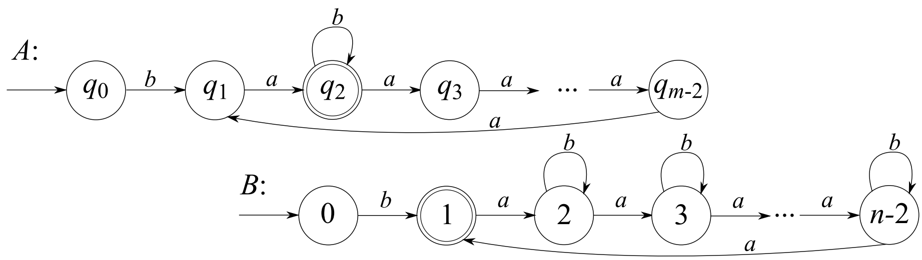


**Fig. 4.** Suffix-free dfa's $A$ and $B$ on binary alphabet

**Theorem 5 (Concatenation: Binary Case).** *Let* $m \geq 4$, $n \geq 3$, *and let* $m-2$ *and* $n-2$ *be relatively prime. There exist binary suffix-free regular languages* $K$ *and* $L$ *with* $\mathrm{sc}(K) = m$ *and* $\mathrm{sc}(L) = n$ *such that* $\mathrm{sc}(KL) \geq (m-1)2^{n-2}$.

*Proof.* Let $K$ and $L$ be the languages accepted by dfa's $A$ and $B$ shown in Fig. 4. By Lemma 1, languages $K$ and $L$ are suffix-free. Construct an nfa for $KL$ from dfa's $A$ and $B$ by adding the transition on $b$ from state $q_2$ to state 1, and by declaring $q_2$ as a rejecting state.

The lengths of the cycles in $A$ and $B$ are $m-2$ and $n-2$, respectively. Since $m-2$ and $n-2$ are relatively prime, there exist integers $y$ and $x$ such that $(m-2)y \equiv 1 \pmod{n-2}$ and $(n-2)x \equiv 1 \pmod{m-2}$.

The initial state of the subset automaton is $\{q_0\}$. We first show that for every subset $X$ of $\{1, 2, \ldots, n-2\}$, state $\{q_2\} \cup X$ is reachable. The proof is by induction on $|X|$. The basis, $|X| = 0$, holds since $\{q_2\}$ is reached from the initial state $\{q_0\}$ by $ba$. Let $X = \{j_1, j_2, \ldots, j_k\}$ with $j_1 < j_2 < \cdots < j_k$ be a subset of $\{1, 2, \ldots, n-2\}$ of size $k$. State $\{q_2\} \cup X$ is reached from state $\{q_2\} \cup \{j_2 - j_1 + 1, \ldots, j_k - j_1 + 1\}$ by $ba^{(m-2)y(j_1-1)}$. Now if $i \in \{1, \ldots, m-2\}$, then state $\{q_i\} \cup X$ is reached from state $\{q_2\} \cup X$ by string $a^{(n-2)x(m-4+i)}$. If $i = m-1$ and $X \neq \{1, \ldots, n-2\}$, then there exists some $z \in \{1, \ldots, n-2\}$ such that $z \notin X$. By string $a^{(m-2)y(n-2-z)+1}$, state $\{q_3\} \cup X'$ is reached from $\{q_2\} \cup X$, where $X'$ is a rotation of $X$ such that $1 \notin X'$. Then from this state, we can reach $\{q_{m-1}\} \cup X'$ by reading $b$. Then we can reach the desired state $\{q_{m-1}\} \cup X$ by reading $a^{z-1}$. This proves the reachability of $(m-1)2^{n-2}$ states.

It remains to show that these states are pairwise distinguishable. Since string $bab$ is accepted by the nfa only from state $q_0$, the initial state $\{q_0\}$ of the subset automaton is distinguishable from any other state. States $\{q_i\} \cup X$ and $\{q_j\} \cup Y$ with $i < j$ are distinguished by $a^{m-i}b$. Finally, states $\{q_i\} \cup X$ and $\{q_i\} \cup Y$ with $X \neq Y$ differ in a state $j$ in $\{1, 2, \ldots, n-2\}$, and so string $a^{n-j-1}$ distinguishes the two states. $\square$

Han and Salomaa [8] proved that $2^{n-2} + 1$ states are sufficient for reversal of suffix-free languages. They met this bound using a ternary alphabet. Theorem 6 shows that this upper bound cannot be met in the binary case.

**Theorem 6 (Reversal).** *Let $L$ be a suffix-free regular language with* $\mathrm{sc}(L) = n$, *where* $n \geq 3$. *Then* $\mathrm{sc}(L^R) \leq 2^{n-2} + 1$. *The bound is tight in the ternary case, but cannot be met in the binary case.*

*Proof.* The bound is tight in the ternary case as shown in [8]. Let us suppose by contradiction that there exists a minimal $n$-state suffix-free dfa $A$ over a binary alphabet such that the minimal dfa for the language $L(A)^R$ has $2^{n-2}+1$ states. Let the set of states of $A$ be $Q$, with the initial state $s$ and the dead state $d$. Construct nfa $A^R$ from the dfa $A$ by reversing all the transitions, and by swapping the role of the initial and final states. The dead state $d$ becomes unreachable in $A^R$, so we can omit it. The subset automaton corresponding to nfa $A^R$, after removing unreachable states, is a minimal dfa for $L(A)^R$ [2].

No subset $X$ of $Q$ such that $s \in X$ and $\{s\} \neq X$ is reachable in the subset automaton [8, Lemma 6]. It follows that the state set of the subset automaton consists of all the subsets of $Q - \{s, d\}$ and state $\{s\}$, that is, $2^{n-2} + 1$ states in total. This means that the set $Z = Q - \{s, d\}$ is reachable in the subset automaton. Since in dfa $A$ there is a transition from state $s$ to some state $q$ in $Q - \{s, d\}$ on some letter $a$, in the subset automaton there is a transition on $a$ from $Z$ to a subset $Y$ such that $s \in Y$. If there are some states $p, p'$ in $Q - \{s, d\}$ such that $p'$ goes to $p$ by $a$ in dfa $A$, then $p' \in Y$. Thus $\{s, p'\} \subseteq Y$. This is a contradiction since such a state cannot be reachable in the subset automaton.

There remains the case when there are no states $p, p'$ in $Q - \{s, d\}$ such that $p'$ goes to $p$ by $a$ in dfa $A$. Then there are just transitions on $b$ among the states in $Q - \{s, d\}$. The determinization of $(n-2)$-state nfa over a unary alphabet requires $e^{\Theta(\sqrt{(n-2)\ln(n-2)})} = o(2^{n-2} + 1)$ states [6]. □

The next theorem provides upper and lower bounds on the state complexity of the reversal of suffix-free languages in the binary case.

**Theorem 7 (Reversal: Binary Case).** *Let $f_2(n)$ be the state complexity of reversal of binary suffix-free languages. Then* $2^{n/2-2}+1 \leq f_2(n) \leq 2^{n-4}+2^{n-3}+1$ *for every integer $n$ with $n \geq 12$.*

*Proof.* Let us prove the upper bound. We will continue our considerations from the proof of previous Theorem 6. Let $K$ be a language over $\{a, b\}$ accepted by an $n$-state suffix-free minimal dfa $A$ with the state set $Q = \{s, d, 1, 2, \ldots n-2\}$, the initial state $s$, and the dead state $d$. Without loss of generality, state $s$ goes to a state $q$ in $\{1, 2, \ldots, n-2\}$ by $a$. Construct nfa $A^R$ as in the previous proof.

First consider the case when there are two states $p$ and $p'$ in $\{1, 2, \ldots, n-2\}$ such that $p$ goes to $p'$ by $a$ in dfa $A$. We have shown in the previous proof that we cannot reach the subset $Q - \{s, d\}$. Moreover, we cannot reach any subset containing both $p$ and $q$ due to the same argument. There are $2^{n-4}$ subsets of $Q - \{s, d\}$ that did not contain neither $p$ nor $q$. And there are $2^{n-3}$ subsets that contain at least one but not both of them. So, including the final state $\{s\}$, there are at most $2^{n-4} + 2^{n-3} + 1$ states in the corresponding subset automaton. If among the states in $\{1, 2, \ldots, n-\ 2\}$, there are just transitions on symbol $b$, then the state complexity of $K^R$ is asymptotically equal to $1 + e^{\sqrt{(n-2)\ln(n-2)}}$.

Now we prove the lower bound. Consider the language $\#L_{m-2}$, where $L_{m-2}$ is the language over $\{a, b\}$ accepted by the $(m-2)$-state Šebej's dfa [12] meeting the upper bound $2^{m-2}$ for reversal. The minimal dfa $B_{\#}$ for $\#L_{m-2}$ has $m$ states. Language $\#L_{m-2}$ is suffix-free. Now, we can construct an automaton $C$ over a binary alphabet by encoding the three alphabet symbols of $B_{\#}$ with two symbols 0 and 1 as follows. For symbol # we use code 00, for symbol $a$ code 10, and for $b$ we use 11. For every state $q$ in $B_{\#}$, we add two special states $q'$ and $q''$, and replace the transitions from $q$ as follows. The transition from $q$ to some $q_{\#}$ by # is replaced with two transitions: The first one goes from $q$ to $q'$ by 0, and the second one from $q'$ to $q_{\#}$ by 0. The transition from $q$ to some $q_a$ by $a$ is replaced with transitions from $q$ to $q''$ by 1 and from $q''$ to $q_a$ by 0. The transition from $q$ to some $q_b$ by $b$ is replaced with transitions from $q$ to $q''$ by 1 and from $q''$ to $q_b$ by 1. All the transitions not defined above go to the dead state. The number of states in $C$ is $3m$. However, there are $m+1$ states that are equivalent to the dead state: For the initial state $s$, state $s''$ goes to the dead state $d$ by both 0 and 1. For all states $q$ except for $s$, $q'$ goes to $d$ by both 0 and 1, and in the case of $d$, also state $d''$ goes to $d$ by 0 and 1. So we can replace all the mentioned special states with $d$. After removing these equivalent states, there remains $2m-1$ states in $C$. Automaton $C$ is deterministic. Let us prove that it is suffix-free. Suppose by contradiction that $C$ accepts a string $w$ and also its proper suffix $v$. Both are of even length, since strings of odd length are not accepted by $C$. So they can be decoded as $w'$ and $v'$, respectively, where $v'$ is a suffix of $w'$ and both are accepted by $B_{\#}$, which is a contradiction.

Now we prove that $C$ accepts the encoded language of $B_{\#}$. If the length of a string $w$ in $\{0,1\}^*$ is odd, then it is rejected in $C$ since it ends in a special state or in $d$. If the length is even, then $w$ can be decoded and it is accepted in $C$ if and only if the decoded $w$ is accepted in $B_{\#}$. We construct nfa $C'$ for language $L(C)^R$ by reversing all the transitions in $C$. In the corresponding subset automaton, we can reach all the $2^{m-2}+1$ states which are pairwise distinguishable, as in the subset automaton for $L(B_{\#})^R$. So if $n$ is the size of the minimal dfa for the encoded language, then $L(C)^R$ requires at least $2^{n/2-2}+1$ states. □

## 4 Conclusions

We gave a characterization of suffix-free nfa's and a sufficient condition on a dfa to accept a suffix-free language. This allowed us to avoid proofs of suffix-freeness of all the languages we have used throughout the paper. Then we investigated the operational state complexity of suffix-free regular languages. We solved completely the case of difference, symmetric difference, and Kleene star since we proved that the general upper bounds for these operations can be met in the binary case.

In the case of concatenation, we provided ternary witness languages. For the binary case, we presented an example that almost meets the upper bound in infinitely many cases. It remains open whether the bound for concatenation can be met in the binary case.

Then we showed that the upper bound for reversal cannot be met in the binary case, and we also gave lower and upper bounds for that case. The exact value of the state complexity of reversal in the binary case remains open.

## References

1. Bordihn, H., Holzer, M., Kutrib, M.: Determinization of finite automata accepting subregular languages. Theoret. Comput. Sci. 410, 3209–3222 (2009)
2. Brzozowski, J.: Canonical regular expressions and minimal state graphs for definite events. In: Mathematical Theory of Automata. MRI Symposia Series, vol. 12, pp. 529–561. Polytechnic Press, Polytechnic Institute of Brooklyn, NY (1962)
3. Brzozowski, J., Jirásková, G., Li, B.: Quotient complexity of ideal languages. In: López-Ortiz, A. (ed.) LATIN 2010. LNCS, vol. 6034, pp. 208–221. Springer, Heidelberg (2010)
4. Brzozowski, J., Jirásková, G., Zou, C.: Quotient complexity of closed languages. In: Ablayev, F., Mayr, E.W. (eds.) CSR 2010. LNCS, vol. 6072, pp. 84–95. Springer, Heidelberg (2010)
5. Câmpeanu, C., Salomaa, K., Culik II, 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)
6. Chrobak, M.: Finite automata and unary languages. Theoret. Comput. Sci. 47, 149–158 (1986); Erratum: Theoret. Comput. Sci. 302, 497–498 (2003)
7. 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. University of Szeged, Hungary (2009)
8. Han, Y.-S., Salomaa, K.: State complexity of basic operations on suffix-free regular languages. Theoret. Comput. Sci. 410, 2537–2548 (2009)
9. Jirásková, G., Olejár, P.: State complexity of intersection and union suffix-free languages and descriptional complexity. In: Bordihn, H., Freund, R., Holzer, M., Kutrib, M., Otto, F. (eds.) NCMA 2009, pp. 151–166. Osterreichische Computer Gesellschaft (2009)
10. Jirásková, G., Masopust, T.: Complexity in union-free regular languages. In: Gao, Y., Lu, H., Seki, S., Yu, S. (eds.) DLT 2010. LNCS, vol. 6224, pp. 255–266. Springer, Heidelberg (2010)
11. Rabin, M., Scott, D.: Finite automata and their decision problems. IBM Res. Develop. 3, 114–129 (1959)
12. Šebej, J.: Reversal of regular languages and state complexity. In: Pardubská, D. (ed.) ITAT 2010, pp. 47–54. P. J. Šafárik University of Košice, Slovakia (2010)
13. Sipser, M.: Introduction to the theory of computation. PWS Publishing Company, Boston (1997)
14. Yu, S.: Regular languages. In: Rozenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages, vol. I, ch. 2, pp. 41–110. Springer, Heidelberg (1997)