Complexity in Union-Free Regular Languages

Galina Jirásková [1], * and Tomáš Masopust [2], **

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

[2] Mathematical Institute, Czech Academy of Sciences
Žižkova 22, 616 62 Brno, Czech Republic
masopust@ipm.cz

Abstract. We continue the investigation of union-free regular languages that are described by regular expressions without the union operation. We also define deterministic union-free languages as languages recognized by one-cycle-free-path deterministic finite automata, and show that they are properly included in the class of union-free languages. We prove that (deterministic) union-freeness of languages does not accelerate regular operations, except for the reversal in the nondeterministic case.

1 Introduction

Regular languages are the simplest languages in the Chomsky hierarchy. They have been intensively investigated due to their practical applications in various areas of computer science, and for their importance in the theory as well.

In recent years, several special subclasses have been deeply examined: finite languages that can be described by expressions without the star operation [6, 7, 32], suffix- and prefix-free languages that are used in codes [12], star-free and locally testable languages, ideal, closed, and convex languages, etc.

Here we continue this research and study union-free regular languages that can be represented by regular expressions without the union operation. Nagy in [26] described one-cycle-free-path nondeterministic finite automata, in which from each state, there is exactly one cycle-free path to the final state. He showed that such automata accept exactly the class of union-free languages. We first complement his results with some closure properties. Then, in Section 3, we investigate the nondeterministic state complexity of operations in the class of union-free languages. Quite surprisingly, we show that all known upper bounds can be reached by union-free languages, except for the reversal, where the tight bound is n instead of $n + 1$. In Section 4, we define deterministic union-free languages as languages accepted by deterministic one-cycle-free-path automata, and show that they are properly included in the class of union-free languages. We study the state complexity of quite a number of regular operations, and prove that deterministic union-freeness of languages does not accelerate any of them.

* Research supported by the VEGA Grant no. 2/0111/09.
** Research supported by the CAS Institutional Research Plan no. AV0Z10190503.

To conclude this section, we mention three more related works. Crvenković, Dolinka, and Ésik [9] investigated algebraic properties of union-free languages. Nagy [25] and Afonin and Golomazov [2] studied union-free decompositions of regular languages.

2 Preliminaries

We assume familiarity with basic concepts of finite automata and regular languages. For all unexplained notions, we refer the reader to [29, 31, 32].

If Σ is a finite alphabet, then Σ^* denotes the set of all strings over the alphabet Σ including the empty string ε. A *language* over an alphabet Σ is any subset of Σ^*. We denote the size of a finite set A by $|A|$ and its power-set by 2^A.

A *nondeterministic finite automaton* (nfa) is a quintuple $M = (Q, \Sigma, \delta, S, F)$, where Q is a finite non-empty set of states, Σ is an input alphabet, S is the set of initial states, F is the set of accepting states, and δ is the transition function that maps $Q \times (\Sigma \cup \{\varepsilon\})$ into 2^Q. The transition function is extended to the domain $2^Q \times \Sigma^*$ in a natural way. The *language accepted by* the nfa M is the set of all strings accepted by M. The automaton M is *deterministic* (dfa) if it has a single initial state, no ε-transitions, and $|\delta(q, a)| = 1$ for all states q in Q and symbols a in Σ. In this case, we usually write $\delta : Q \times \Sigma \to Q$.

A language is *regular* if there exists an nfa (or a dfa) accepting the language. The *state complexity* of a regular language L, denoted by $\mathrm{sc}(L)$, is the smallest number of states in any dfa accepting the language L. The *nondeterministic state complexity* of a regular language L, $\mathrm{nsc}(L)$, is defined as the smallest number of states in any ε-free nfa that accepts L and has a single initial state.

A path from state p to state q in an nfa/dfa M is a sequence $p_0 a_1 p_1 a_2 \cdots a_n p_n$, where $p_0 = p$, $p_n = q$, and $p_i \in \delta(p_{i-1}, a_i)$ for $i = 1, 2, \ldots, n$. The path is called *accepting cycle-free* if p_n is an accepting state, and $p_i \neq p_j$ whenever $i \neq j$. An nfa/dfa is a *one-cycle-free-path* (1cfp) nfa/dfa if there is a unique accepting cycle-free path from each of its states.

A *regular expression* over an alphabet Σ is defined inductively as follows: $\emptyset$, ε, and a, for a in Σ, are regular expressions. If r and t are regular expressions, then also $(s \cup t)$, $(s \cdot t)$, and $(s)^*$ are regular expressions. A regular expression is *union-free* if no symbol $\cup$ occurs in it. A regular language is *union-free* if there exists a union-free regular expression describing the language.

Let K and L be languages over Σ. We denote by $K \cap L$, $K \cup L$, $K - L$, $K \oplus L$ the intersection, union, difference, and symmetric difference of K and L, respectively. To denote complement, Kleene star, and reversal of L, we use L^c, L^*, and L^R. The left and right quotient of a language L with respect to a string w is the set $w \backslash L = \{x \mid wx \in L\}$ and $L/w = \{x \mid xw \in L\}$, respectively. The cyclic shift of a language L is defined as $L^{shift} = \{uv \mid vu \in L\}$. The *shuffle* of two languages is $K \sqcup\!\sqcup L = \{u_1 v_1 u_2 v_2 \cdots u_m v_m \mid m \geqslant 1,\ u_i, v_i \in \Sigma^*,\ u_1 \cdots u_m \in K,\ v_1 \cdots v_m \in L\}$. For the definition of positional addition, $K + L$, we refer to [18]: informally, strings are considered as numbers encoded in a $|\Sigma|$-adic system, and automata read their inputs from the least significant digit.

3 Union-Free Regular Languages

A regular language is union-free if it can be described by a union-free regular expression. Nagy in [26] proved that the class of union-free regular languages coincides with the class of languages recognized by one-cycle-free-path nfa's. He also showed that union-free languages are closed under concatenation, Kleene-star, and substitution by a union-free language. Using an observation that the shortest string of a union-free language is unique, he proved not closeness under union, complementation, intersection, and substitution by a regular language. Our first result complements the closure properties.

Theorem 1 (Closure Properties). *The class of union-free regular languages is closed under reversal, but not closed under cyclic shift, shuffle, symmetric difference, difference, left and right quotients, and positional addition.*

Proof. We prove the closeness under reversal by induction on the structure of a regular expression. If r is $\emptyset$, or ε, or a, then the reversal is described by the same expression. If $r = st$, or $r = s^*$, then the reversal is $L(t)^R L(s)^R$ or $(L(s)^R)^*$, respectively, which are union-free due to closeness under concatenation and star.

To prove the non-closure properties, we give union-free languages such that the shortest string in the resulting language is always of length two, and we show that there are at least two such strings in all cases: $\{ab\}^{shift} = \{a\} \sqcup\!\sqcup \{b\} = \{ab\} \oplus \{ba\} = \{ab, ba\}$; $a(b \cup c)^* - a^* = \{ab, ac, \ldots\}$; $g \backslash (ge \cup gf)^* b = \{eb, fb, \ldots\}$ and $a(eb \cup fb)^*/b = \{ae, af, \ldots\}$; $88^* + 33^* = \{11, 19, \ldots\}$. As the shortest strings are not unique, the resulting languages are not union-free. $\square$

The subset construction assures that every nfa of n states can be simulated by a dfa of at most 2^n states. The worst case binary examples are known for a long time [20, 22, 24]. In addition, Domaratzki et al. [10] have shown that there are at least 2^{n-2} distinct binary languages recognized by nfa's of n states that require 2^n deterministic states. However, none of the above mentioned automata is one-cycle-free-path nfa. The following theorem shows that the bound 2^n is tight in the class of union-free regular languages as well.

Theorem 2 (NFA to DFA Conversion). *For every positive integer n, there exists a binary one-cycle-free-path nfa of n states whose equivalent minimal dfa has 2^n states.*

Proof. Consider the binary 1cfp nfa with states $0, 1, \ldots, n-1$, of which 0 is the initial state, and $n-1$ is the sole accepting state. By a, each state i goes to $\{i+1\}$, except for state $n-1$ which goes to the empty set. By b, each state i goes to $\{0, i\}$. Let us show that the corresponding subset automaton has 2^n reachable and pairwise inequivalent states. Each singleton $\{i\}$ is reached from the initial state $\{0\}$ by a^i, and the empty set is reached by a^n. Each set $\{i_1, i_2, \ldots, i_k\}$, where $0 \leqslant i_1 < i_2 < \cdots < i_k \leqslant n-1$, of size k $(2 \leqslant k \leqslant n)$ is reached from the set $\{i_2 - i_1, i_3 - i_1, \ldots, i_k - i_1\}$ of size $k-1$ by the string ba^{i_1}. This proves the reachability of all subsets. For inequivalence, notice that the string a^{n-1-i} is accepted by the nfa only from state i. Two different subsets must differ in a state i, and so the string a^{n-1-i} distinguishes the two subsets. $\square$

We next study the nondeterministic state complexity of regular operations in the class of union-free languages. Quite surprisingly, all upper bounds can be reached by union-free languages, except for the reversal where the upper bound is n instead of $n+1$. To prove the results we use a fooling set lower-bound technique [3–5, 11, 14].

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 (1) for all i, the string $x_i y_i$ is in the language L, and (2) if $i \neq j$, then at least one of the strings $x_i y_j$ and $x_j y_i$ is not in the language L.

It is well-known that the size of a fooling set for a regular language provides a lower bound on the number of states in any nfa for this language. The argument is simple. We can fix accepting computations of any nfa on strings $x_i y_i$. The states on these computations reached after reading x_i must be pairwise distinct, otherwise the nfa would accept both $x_i y_j$ and $x_j y_i$ for two distinct pairs.

The next lemma shows that sometimes one more state is necessary. The lemma can be used to simplify some proofs from the literature, for example, the results on union, reversal, and cyclic shift of nfa languages.

Lemma 1. *Let L be a regular language. Let $\mathcal{A}$ and $\mathcal{B}$ be sets of pairs of strings and let u and v be two strings such that $\mathcal{A} \cup \mathcal{B}$, $\mathcal{A} \cup \{(\varepsilon, u)\}$, and $\mathcal{B} \cup \{(\varepsilon, v)\}$ are fooling sets for L. Then every nfa for L has at least $|\mathcal{A}| + |\mathcal{B}| + 1$ states.*

Proof. Consider an nfa for L, and let $\mathcal{A} = \{(x_i, y_i) \mid i = 1, 2, \ldots, m\}$ and $\mathcal{B} = \{(x_{m+j}, y_{m+j}) \mid j = 1, 2, \ldots, n\}$. Since the strings $x_k y_k$ are in L, we can fix an accepting computation of the nfa on each string $x_k y_k$. Let p_k be the state on this computation that is reached after reading x_k. Since $\mathcal{A} \cup \mathcal{B}$ is a fooling set for L, the states $p_1, p_2, \ldots, p_{m+n}$ must be pairwise distinct. Since $\mathcal{A} \cup \{(\varepsilon, u)\}$ is a fooling set, the initial state must be distinct from all states $p_1, p_2, \ldots, p_m$. Since $\mathcal{B} \cup \{(\varepsilon, v)\}$ is a fooling set, the initial state must also be distinct from all states $p_{m+1}, p_{m+2}, \ldots, p_{m+n}$. Thus the nfa has at least $m + n + 1$ states. □

Theorem 3 (Nondeterministic State Complexity). *Let K and L be union-free regular languages over an alphabet Σ accepted by an m-state and an n-state one-cycle-free-path nfa, respectively. Then*

1. $\mathrm{nsc}(K \cup L) \leqslant m + n + 1$, *and the bound is tight if* $|\Sigma| \geqslant 2$;
2. $\mathrm{nsc}(K \cap L) \leqslant mn$, *and the bound is tight if* $|\Sigma| \geqslant 2$;
3. $\mathrm{nsc}(KL) \leqslant m + n$, *and the bound is tight if* $|\Sigma| \geqslant 2$;
4. $\mathrm{nsc}(K \sqcup L) \leqslant mn$, *and the bound is tight if* $|\Sigma| \geqslant 2$;
5. $\mathrm{nsc}(K + L) \leqslant 2mn + 2m + 2n + 1$, *and the bound is tight if* $|\Sigma| \geqslant 6$;
6. $\mathrm{nsc}(L^2) \leqslant 2n$, *and the bound is tight if* $|\Sigma| \geqslant 2$;
7. $\mathrm{nsc}(L^c) \leqslant 2^n$, *and the bound is tight if* $|\Sigma| \geqslant 3$;
8. $\mathrm{nsc}(L^R) \leqslant n$, *and the bound is tight if* $|\Sigma| \geqslant 1$;
9. $\mathrm{nsc}(L^*) \leqslant n + 1$, *and the bound is tight if* $|\Sigma| \geqslant 1$;
10. $\mathrm{nsc}(L^{shift}) \leqslant 2n^2 + 1$, *and the bound is tight if* $|\Sigma| \geqslant 2$;

Proof. 1. To get an nfa for the union, we add a new initial state that goes by the empty string to the initial states of the given automata. For tightness, consider binary union-free languages $(a^m)^*$ and $(b^n)^*$ [13], and the following sets of pairs of strings: $\mathcal{A} = \{(a^i, a^{m-i}) \mid i = 1, 2, \ldots, m-1\} \cup \{(a^m, a^m)\}$ and $\mathcal{B} = \{(b^j, b^{n-j}) \mid j = 1, 2, \ldots, n-1\} \cup \{(b^n, b^n)\}$. Let $L = (a^m)^* \cup (b^n)^*$, and let us show that the set $\mathcal{A} \cup \mathcal{B}$ is a fooling set for the language L. The concatenation of the first and the second part of each pair results in a string in $\{a^m, a^{2m}, b^n, b^{2n}\}$, and so is in the language L. Next, the concatenation of the first part of a pair and the second part of another pair results in a string in $\{a^r, a^{m+r}, b^s, b^{n+s}, a^r b^s, b^s a^r, a^m b^n, b^n a^m \mid 0 < r < m, 0 < s < n\}$, and so is not in L. Finally, both sets $\mathcal{A} \cup \{(\varepsilon, b^n)\}$ and $\mathcal{B} \cup \{(\varepsilon, a^m)\}$ are fooling sets for L as well. By Lemma 1, every nfa for L has at least $m + n + 1$ states.

2. Standard cross-product construction provides the upper bound mn on the intersection. To prove that the bound is tight consider binary 1cfp nfa's that count the number of a's modulo m and the number of b's modulo n, respectively. Since the set $\{(a^i b^j, a^{m-i} b^{n-j}) \mid 0 \leqslant i \leqslant m-1,\ 0 \leqslant j \leqslant n-1\}$ is a fooling set of size mn for the intersection of the languages accepted by the two automata.

3. To get an nfa for the concatenation from two given nfa's, we only need to add an ε-transition from all final states in the first automaton to the initial state in the second automaton. For tightness, consider languages $(a^m)^*$ and $(b^n)^*$. The set $\{(a^i, a^{m-i} b^n) \mid i = 0, 1, \ldots, m-1\} \cup \{(a^m b^j, b^{n-j}) \mid j = 1, 2, \ldots, n\}$ is a fooling set of size $m + n$ for the concatenation of the two languages.

4. The state set of an nfa for the shuffle is the product of the state sets of given nfa's, and its transition function δ is defined using transitions functions δ_A and δ_B of the given automata by $\delta((p, q), a) = \{(\delta_A(p, a), q), (p, \delta_B(q, a))\}$ [8]. This gives the upper bound mn. The bound is reached by the shuffle of the languages $(a^m)^*$ and $(b^n)^*$ since the set $\{(a^i b^j, a^{m-i} b^{n-j}) \mid 0 \leqslant i \leqslant m-1,\ 0 \leqslant j \leqslant n-1\}$ is a fooling set of size mn for the shuffle.

5. The nfa for positional addition in [18] consists of $2mn + 2m + 2n + 1$ states, and it is shown here that the bound is reached by the positional addition of union-free languages $((1^* 5)^m)^*$ and $((2^* 5)^n)^*$ over the alphabet $\{0, 1, 2, 3, 4, 5\}$.

6. Since L^2 is the concatenation of the language L with itself, the upper bound $2n$ follows from part 3. To prove tightness consider the 1cfp nfa shown in Fig. 1. Construct an nfa with the state set $Q = \{p_0, p_1, \ldots, p_{n-1}\} \cup \{q_0, q_1, \ldots, q_{n-1}\}$

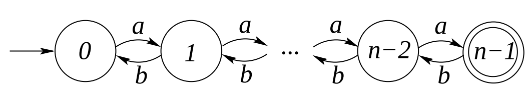

Fig. 1. The one-cycle-free nfa reaching the bound $2n$ on square.

for the language L^2 from two copies of the nfa for L by adding an ε-transition from the final state in the first copy to the initial state in the second copy. The initial state of the resulting nfa is p_0, the only final state is q_{n-1}. For each state s in Q, define strings x_s and y_s as follows (notice that for each state s, the initial state p_0 goes to s by x_s, and each s goes to the accepting state q_{n-1} by y_s):

$$x_s = \begin{cases} a^i & \text{if } s = p_i, \\ a^{2n-2}b^{n-1-i} & \text{if } s = q_i, \end{cases} \qquad y_s = \begin{cases} a^{2n-2-i} & \text{if } s = p_i \text{ and } i \neq n-1, \\ b^{n-1}a^{2n-2} & \text{if } s = p_{n-1}, \\ a^{n-1-i} & \text{if } s = q_i. \end{cases}$$

Then the set $\{(x_s, y_s) \mid s \in Q\}$ is a fooling set for the language L^2 of size $2n$.

7. After applying subset construction to a given nfa and interchanging the accepting and rejecting states, we get an nfa (even a dfa) of at most 2^n states for the complement of the language recognized by the given nfa. The bound has been proved to be tight for a growing alphabet in [28], for a four-letter alphabet in [5], and for a binary alphabet in [16]. However, the binary witness nfa's in [16] are not 1cfp. We prove the tightness of the bound also in the class of 1cfp automata. To this aim consider the ternary language L recognized by the 1cfp nfa in Fig. 2; denote the state set $\{0, 1, \ldots, n-1\}$ by Q. By c, state $n-1$ goes

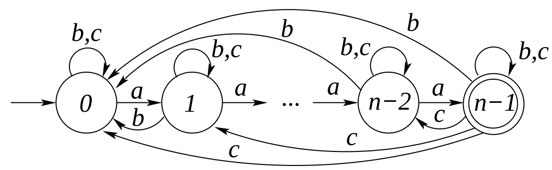

Fig. 2. The one-cycle-free nfa reaching the bound 2^n on complement.

to $\{0, 1, \ldots, n-1\}$, and each other state i goes to $\{i\}$. Transitions by a and b are the same as in the automaton in the proof of Theorem 2. Therefore, in the corresponding subset automaton, each subset S of the state set Q is reached from the initial state $\{0\}$ by a string x_S in $\{a, b\}^*$. We are now going to define strings y_S so that the set $\{(x_S, y_S) \mid S \subseteq Q\}$ would be a fooling set for L^c.

Let S be a subset of Q. If $S = \{0, 1, \ldots, n-2\}$, let $y_S = c$, otherwise let $y_S = y_1 y_2 \cdots y_n$, where for each i in Q, $y_{n-i} = a$ if $i \in S$, and $y_{n-i} = ca$ if $i \notin S$. Then the set $\{(x_S, y_S) \mid S \subseteq Q\}$ is a fooling set for the language L^c of size 2^n.

8. To get an n-state nfa for the reversal of a language accepted by an n-state 1cfp nfa, we reverse all transitions, make the initial state final, and (the only) final state the initial. The unary union-free language a^{n-1} reaches the bound.

9. The standard construction of an nfa for the Kleene star that adds a new initial (and accepting) state connected through an ε-transition to the initial state of the given nfa as well as ε-transitions from each final state to the initial state, provides the upper bound $n+1$. For tightness, consider the union-free language $a^{n-1}(a^n)^*$. The set $\{(\varepsilon, \varepsilon)\} \cup \{(a^i, a^{n-1-i}) \mid i = 1, 2, \ldots, n-2\} \cup \{(a^{n-1}, a^n), (a^n, a^{n-1})\}$ is a fooling set of size $n+1$ for the star of this language.

10. The nfa for cyclic shift in consists of $2n^2 + 1$ states, and the one-cycle-free-free nfa in Fig. 1 reaches the bound [17]. To prove the result, a fooling set of size $2n^2$ is described in [17], and then Lemma 1 is used to show that one more state is necessary. □

4 Deterministic Union-Free Regular Languages

We now turn our attention to deterministic union-free languages, that is, to languages that are recognized by one-cycle-free-path deterministic finite automata. We first show that deterministic union-free languages are properly included in the class of union-free languages. Then we study the state complexity of regular operations in the class of deterministic union-free languages.

Theorem 4 (1cfp DFAs vs. 1cfp NFAs). *The class of deterministic union-free languages is a proper subclass of the class of union-free regular languages.*

Proof. Let $k \geqslant 3$. We show that there exists a unary union-free regular language such that every dfa for this language has at least k final states, and so the language is not deterministic union-free. Set $n = k(k-1)/2$.

Define a unary $2n$-state dfa with states $0, 1, \ldots, 2n-1$, of which 0 is the initial state. The set of final states is $\{0, n, n+(k-1), n+(k-1)+(k-2), \ldots, 2n-1\}$. Each state i goes by a to state $i+1$, except for state $2n-1$ that goes to itself. Let L be a language recognized by this dfa. Since

$$n + (k-1) + (k-2) + \cdots + (k-(k-2)) = 2n-1,$$

there are $k-2$ final states greater than n, and so the dfa has k final states. Moreover, state $2n-2$ is not final. We now show that the automaton is minimal. Let i and j be two states with $i < j$. Then there exists an integer m such that by the string a^m, state j goes to state $2n-1$, while state i goes to state $2n-2$. Since state $2n-1$ is final and state $2n-2$ is not, the states i and j are inequivalent and the dfa is minimal. It turns out that every dfa for the language L must have at least k final states, and so the language L is not deterministic union-free.

To prove that the language L is union-free, we describe a 1cfp nfa for L. The only initial and final state of the nfa is state 0. Next, construct $k+n$ cycles that are pairwise disjoint, except for state 0. The length of the cycles is consequently $n, n+(k-1), n+(k-1)+(k-2), \ldots, 2n-1$, and then $2n, 2n+1, \ldots, 3n-1$. The automaton is 1cfp nfa, accepts all strings in L of length less that $2n$, as well as all strings of length at least $2n$, but no other strings since going through more than one cycle results in a string of length at least $2n$. □

The next theorem shows that deterministic union-freeness of languages does not accelerate basic regular operations. This contrasts with the results in previously studied subclasses of regular languages such as finite, unary, prefix-, suffix-, factor-, subword-free (or closed, or convex) etc. In the case of intersection and square, the known witness languages are deterministic union-free [33, 27]. Slightly changed Maslov's automata [21] provide lower bounds for star and concatenation, while a modification of the hardest dfa in [17] gives a lower bound for cyclic shift. In the case of reversal, the paper [30] claims that there is a binary n-state dfa language whose reversal requires 2^n deterministic states. Although the witness automaton is one-cycle-free-path dfa, the result cannot be used because the proof is not correct. If $n = 8$, then the resulting dfa has only 252 states instead of 256, as the reader can verify using a software, for example, in [1].

Theorem 5 (State Complexity). *Let K and L be deterministic union-free regular languages over an alphabet Σ accepted by an m-state and an n-state one-cycle-free-path dfa, respectively. Then*

1. $\mathrm{sc}(K \cup L) \leqslant mn$*, and the bound is tight if* $|\Sigma| \geqslant 2$*;*
2. $\mathrm{sc}(K \cap L) \leqslant mn$*, and the bound is tight if* $|\Sigma| \geqslant 2$*;*
3. $\mathrm{sc}(K - L) \leqslant mn$*, and the bound is tight if* $|\Sigma| \geqslant 2$*;*
4. $\mathrm{sc}(K \oplus L) \leqslant mn$*, and the bound is tight if* $|\Sigma| \geqslant 2$*;*
5. $\mathrm{sc}(KL) \leqslant m2^n - 2^{n-1}$ $(m \geqslant 2, n \geqslant 3)$*, and the bound is tight if* $|\Sigma| \geqslant 2$*;*
6. $\mathrm{sc}(L^2) \leqslant n2^n - 2^{n-1}$*, and the bound is tight if* $|\Sigma| \geqslant 2$*;*
7. $\mathrm{sc}(L^c) \leqslant n$*, and the bound is tight if* $|\Sigma| \geqslant 1$*;*
8. $\mathrm{sc}(L^*) \leqslant 2^{n-1} + 2^{n-2}$ $(n \geqslant 2)$*, and the bound is tight if* $|\Sigma| \geqslant 2$*;*
9. $\mathrm{sc}(L^R) \leqslant 2^n$ $(n \geqslant 2)$*, and the bound is tight if* $|\Sigma| \geqslant 3$*;*
10. $\mathrm{sc}(L^{shift}) \leqslant 2^{n^2+n\log n}$*. The bound* $2^{n^2+n\log n-5n}$ *can be reached if* $|\Sigma| \geqslant 4$.

Proof. 1.-4. The cross-product construction gives the upper bound mn. For all four operations, the bound is reached by deterministic union-free binary languages $((b^*a)^m)^*$ and $((a^*b)^n)^*$: the strings a^ib^j with $0 \leqslant i \leqslant m-1$ and $0 \leqslant j \leqslant n-1$ are pairwise inequivalent in the right-invariant congruence defined by the intersection (union, difference, symmetric difference, respectively).

5. The upper bound is $m2^n - 2^{n-1}$ [21, 33]. Notice that neither the ternary witness automata in [33] nor binary witness automata in [15] are 1cfp dfa's. However, Maslov [21] claimed the result for two binary languages accepted by automata, the first of which is 1cfp dfa, while the second one can be made to be 1cfp dfa by changing its accepting state from $n-1$ to $n-2$. Since no proof is provided in [21], we recall the two automata and show that they reach the upper bound. Consider languages accepted by the 1cfp dfa's shown in Fig. 3. Construct an nfa for the concatenation of the two languages from these dfa's

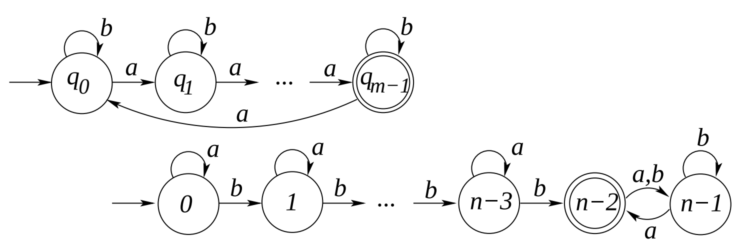

Fig. 3. The one-cycle-free-path dfa's reaching the bound $m2^n - 2^{n-1}$ on concatenation.

by adding an ε-transition from state q_{m-1} to state 0. The initial state of the nfa is state q_0, the sole accepting state is $n-2$. We first prove by induction on the size of subsets that each set $\{q_i\} \cup S$, where $0 \leqslant i \leqslant m-2$ and S is a subset of $\{0, 1, \ldots, n-1\}$, as well as each set $\{q_{m-1}\} \cup T$, where T is a subset of $\{0, 1, \ldots, n-1\}$ containing state 0, is reachable. Each singleton $\{q_i\}$ with $i \leqslant m-2$ is reached from the initial state $\{q_0\}$ by a^i. Assume the reachability of all appropriate sets of size k. Let $S = \{q_i, j_1, j_2, \ldots, j_k\}$ be a subset of size $k+1$. First, let $i = m-1$, and so $j_1 = 0$. Since the symbol a is a permutation symbol in the second dfa, we can use $j \ominus r$ to denote the state that goes to state j by a^r. Consider the set $S' = \{q_{m-2}, j_2 \ominus 1, \ldots, j_k \ominus 1\}$ of size k. The set S' is reachable by the induction hypothesis, and since S' goes to S by a, the set S is reachable as well. Now let $i \leqslant m-2$ and $j_1 = 0$. Then the set S is reached from the set $\{q_{m-1}, 0, j_2 \ominus (i+1), \ldots, j_k \ominus (i+1)\}$ by a^{i+1}. Finally, if $i \leqslant m-2$ and $j_1 > 0$, then the set S is reached from the set $\{q_i, 0, j_2 - j_1, j_3 - j_1, \ldots, j_k - j_1\}$ by b^{j_1}. This concludes the proof of reachability. Now let $\{q_i\} \cup S$ and $\{q_j\} \cup T$ be two different reachable sets. If $i < j$, then the string $ba^{m-j-1}b^{n-2}$ distinguishes the two subsets. If $i = j$, then S and T differ in a state j, and moreover, $j > 0$ if $i = m-1$. Then either the string b^{n-j-2} (if $j \leqslant n-3$), or the empty string (if $j = n-2$), or the string a (if $j = n-1$) distinguishes the two subsets.

6. The upper bound follows from the upper bound on concatenation, and, as shown in [27], is reached by the binary language recognized by the 1cfp dfa with states $0, 1, \ldots, n-1$, of which 0 is the initial state, and $n-1$ is the sole accepting state; by a, each state i goes to state $i+1 \bmod n$, and by b, each state i goes to itself except for state 1 that goes to state 0 by b.

7. To get a dfa for the complement we only need to exchange the accepting and rejecting states. The bound is reached by the language $(a^n)^*$.

8. The upper bound is $2^{n-1} + 2^{n-2}$ [33]. The witness language in [33] is not deterministic union-free. However, Maslov [21] provides deterministic union-free witness example shown in Fig. 4. Since there is no proof in [21], we give it here. Construct an nfa for the star of the language accepted by the 1cfp dfa in Fig. 4

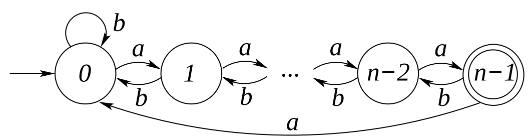

Fig. 4. The one-cycle-free-path dfa reaching the bound $2^{n-1} + 2^{n-2}$ on star.

by adding a new initial and accepting state q_0 that goes to state 1 by a and to state 0 by b, and by adding the transition by a from state $n-2$ to state 0. The initial state $\{q_0\}$ and all singletons $\{i\}$ are reachable. Assume that all subsets of size $k-1$ containing state 0, or containing neither 0 nor $n-1$ are reachable. Let $S = \{i_1, i_2, \ldots, i_k\}$ be a subset of size k with $0 \leqslant i_1 < i_2 < \cdots < i_k \leqslant n-1$ (and if $i_1 > 0$ then $i_k < n-1$). First, let $i_1 = 0$. Then the set S is reached from the set $\{i_2 + (n-1) - i_k - 1, i_3 + (n-1) - i_k - 1, \ldots, i_{k-1} + (n-1) - i_k - 1, n-2\}$ of size $k-1$, containing neither 0 nor $n-1$, by the string ab^{n-1-i_k}. Now let $i_1 > 0$. Then $i_k < n-1$, and the set S is reached from the set $\{0, i_2 - i_1, i_3 - i_1, \ldots, i_k - i_1\}$, which contains state 0, by a. To prove inequivalence notice that the initial (and accepting) state $\{q_0\}$ cannot be equivalent to any state not containing state $n-1$. However, the string a^n is accepted by the nfa from state $n-1$ but not from state q_0. Two different subsets of the state set of the given dfa differ in a state i, and the string a^{n-1-i} distinguishes the two subsets.

9. The reversal of a dfa language is accepted by the nfa obtained from the given dfa by reversing all transitions, making all accepting states initial, and the initial state accepting. The subset construction gives a dfa of at most 2^n states. As pointed out by Mirkin [23], the Lupanov's ternary worst-case example for nfa-to-dfa conversion in [20] is, in fact, a reversed dfa. Leiss [19] presented a ternary and a binary dfa's that reach the the upper bound. Since none of these automata is 1cfp dfa, let us consider the 1cfp dfa shown in Fig. 5. Construct

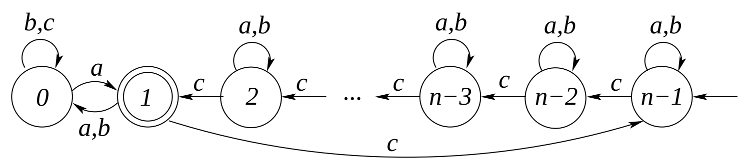

Fig. 5. The one-cycle-free-path dfa reaching the bound 2^n on reversal.

the reversed nfa. Notice that in this nfa each state i goes to state $(i+1) \bmod n$ by ca. It turns out that in the subset automaton, each subset not containing state 0 is reached from a subset containing state 0 by a string in $(ca)^*$. Let us show by induction on the size of subsets that each subset of the state set $\{0,1,\dots,n-1\}$ containing state 0 is reachable in the subset automaton. The singleton $\{0\}$ is reached from the initial state $\{1\}$ of the subset automaton by a. The subset $\{0,i_1,i_2,\dots,i_k\}$, where $1 \leqslant i_1 < i_2 < \cdots < i_k \leqslant n-1$, of size $k+1$ is reached from the set $\{0, i_2-i_1+1, i_3-i_1+1,\dots,i_k-i_1+1\}$ of size k by the string bc^{i_1-1}. Finally, the empty set is reached from state $\{1\}$ by b. For inequivalence, notice that the string c^{n-1-i} is accepted by the nfa only from state i for $i=1,2,\dots,n-1$, and the string ac^{n-2} only from state 0.

10. The upper bound is from [21, 17]. The work [17] proves the lower bound $2^{n^2+n\log n-5n}$ for the language recognized by the dfa over the alphabet $\{a,b,c,d\}$ with states $0,1,\dots,n-1$, of which 0 is the initial state and $n-1$ is the sole accepting state, and transitions are defined as follows: By a, states 0 and $n-1$ go to itself and there is a circle $(1,2,\dots,n-2)$; by b, state 0 goes to itself and there is a circle $(1,2,\dots,n-1)$; by c, all states go to itself except for state 0 that goes to 1 and state 1 that goes to 0; by d, all states go to state 0 except for state $n-1$ that goes to state 1. This automaton is not one-cycle-free-path dfa. Therefore, let us change transitions on symbol b so that in a new dfa by b, all states go to itself except for state $n-2$ that goes to $n-1$ and state $n-1$ that goes to $n-2$. The resulting automaton is a 1cfp dfa, and moreover, the transitions by old symbol b are now implemented by the string ba. It turns out that the proof in [17] works for the new 1cfp dfa if we replace all occurrences of b in the proof by the string ba. □

5 Conclusions

We investigated union-free regular languages that can be described by regular expressions without the union operation. Using known results of Nagy [26] on characterization of automata accepting those languages, we proved some closure properties, and studied the nondeterministic state complexity of regular operations. We showed that all known upper bounds can be reached by union-free languages, except for the reversal, where the tight bound is n instead of $n+1$. We also defined deterministic union-free languages as languages recognized by deterministic one-cycle-free-path automata, and proved that they are properly included in the class of union-free languages. We examined the state complexity of quite a number of regular operations, and showed that deterministic union-freeness of languages accelerates none of them. This contrasts with the results on complexity of operations in previously studied subclasses of regular languages.

Some questions remain open. We conjecture that for the difference of two union-free languages, nfa's need $m2^n$ states, and we do not now the result on the shuffle of deterministic union-free languages. A description of deterministic union-free regular languages in terms of regular expressions or grammars, as well as the case of unary union-free languages, is of interest too.

References

1. JFLAP, http://www.jflap.org/
2. Afonin, S., Golomazov, D.: Minimal union-free decompositions of regular languages. In: Dediu, A.H., Ionescu, A.M., Martín-Vide, C. (eds.) LATA 2009. LNCS, vol. 5457, pp. 83–92. Springer, Heidelberg (2009)
3. Aho, A.V., Ullman, J.D., Yannakakis, M.: On notions of information transfer in VLSI circuits. In: Proc. 15th Annual ACM Symposium on the Theory of Computing, pp. 133–139 (1983)
4. Birget, J.-C.: Intersection and union of regular languages and state complexity. Inform. Process. Lett. 43, 185–190 (1992)
5. Birget, J.-C.: Partial orders on words, minimal elements of regular languages, and state complexity. Theoret. Comput. Sci. 119, 267–291 (1993); Erratum: Available at `http://clam.rutgers.edu/~birget/poWordsERR.ps`
6. Bordihn, H., Holzer, M., Kutrib, M.: Determination of finite automata accepting subregular languages. Theoret. Comput. Sci. 410, 3209–3222 (2009)
7. Câmpeanu, C., Culik II, K., Salomaa, K., Yu. S.: State complexity of basic operations on finite languages. In: Boldt, O., Jürgensen, H. (eds.) WIA 1999. LNCS, vol. 2214, pp. 60–70. Springer, Heidelberg (2001)
8. Câmpeanu, C., Salomaa, K., Yu, S.: Tight lower bound for the state complexity of shuffle of regular languages. J. Autom. Lang. Comb. 7, 303–310 (2002)
9. Crvenković, S., Dolinka, I., Ésik, Z.: On equations for union-free regular languages. Inform. and Comput. 164, 152–172 (2001)
10. Domaratzki, M., Kisman, D., Shallit, J.: On the number of distinct languages accepted by finite automata with n states. J. Autom. Lang. Comb. 7, 469–486 (2002)
11. Glaister, I., Shallit, J.: A lower bound technique for the size of nondeterministic finite automata. Inform. Process. Lett. 59, 75–77 (1996)

12. Han, Y.-S., Salomaa, K.: State complexity of basic operations on suffix-free regular languages. Theoret. Comput. Sci. 410, 2537–2548 (2009)
13. Holzer, M., Kutrib, M.: Nondeterministic descriptional complexity of regular languages. Internat. J. Found. Comput. Sci. 14, 1087–1102 (2003)
14. Hromkovič, J.: Communication complexity and parallel computing. Springer, Heidelberg (1997)
15. Jirásek, J., Jirásková, G., Szabari, A.: State complexity of concatenation and complementation. Internat. J. Found. Comput. Sci. 16, 511–529 (2005)
16. Jirásková, G.: State complexity of some operations on binary regular languages. Theoret. Comput. Sci. 330, 287–298 (2005)
17. Jirásková, G., Okhotin, A.: State complexity of cyclic shift. Theor. Inform. Appl. 42, 335–360 (2008)
18. Jirásková, G., Okhotin, A.: Nondeterministic state complexity of positional addition. In: Dassow, J., Pighizzini, G., Truthe, B. (eds.) 11th International Workshop on Descriptional Complexity of Formal Systems, pp. 199–210. Otto-von-Guericke-Universität, Magdeburg, Germany (2009)
19. Leiss, L.: Succint representation of regular languages by boolean automata. Theoret. Comput. Sci. 13, 323–330 (1981)
20. Lupanov, O.B.: Über den Vergleich zweier Typen endlicher Quellen. Probl. Kybernetik 6, 328–335 (1966); translation from Probl. Kibernetiki 9, 321–326 (1963)
21. Maslov, A.N.: Estimates of the number of states of finite automata. Sov. Math., Dokl. 11, 1373–1375 (1970); translation from Dokl. Akad. Nauk SSSR 194, 1266–1268 (1970)
22. Meyer, A.R., Fischer, M.J.: Economy of description by automata, grammars, and formal systems. In: Proc. 12th IEEE Symp. on Switching and Automata Theory, pp. 188-191 (1971)
23. Mirkin, B.G.: On dual automata. Kibernetika 1, 7–10 (1966)
24. Moore, F.R.: On the bounds for state-set size in the proofs of equivalence between deterministic, nondeterministic, and two-way finite automata. IEEE Trans. Comput. 20, 1211–1219 (1971)
25. Nagy, B.: A normal form for regular expressions. In: Calude, C.S., Calude, E., Dinneen, M.J.: (eds.) CDMTCS-252 report, supplemental material for DLT'04, pp. 53–62. University of Auckland, New Zealand (2004)
26. Nagy, B.: Union-free regular languages and 1-cycle-free-path automata. Publ. Math. Debrecen 68, 183–197 (2006)
27. Rampersad, N.: The state complexity of L^2 and L^k. Inform. Process. Lett. 98, 231–234 (2006)
28. Sakoda, W.J., Sipser, M.: Nondeterminism and the size of two-way finite automata. In: Proc. 10th Annual ACM Symposium on the Theory of Computing, pp. 275–286 (1978)
29. Salomaa, A.: Formal languages. Academic Press, New York (1973)
30. Salomaa, A., Wood, D., Yu, S.: On the state complexity of reversals of regular languages. Theoret. Comput. Sci. 320, 315–329 (2004)
31. Sipser, M.: Introduction to the theory of computation. PWS Publishing Company, Boston (1997)
32. Yu, S.: Chapter 2: Regular languages. In: Rozenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages – Vol. I, pp. 41–110. Springer, Heidelberg (1997)
33. Yu, S., Zhuang, Q., Salomaa, K.: The state complexities of some basic operations on regular languages. Theoret. Comput. Sci. 125, 315–328 (1994)