Acta Cybernetica **21** (2014) 507–527.

Quotient Complexity of Bifix-, Factor-, and Subword-free Regular Languages*

Janusz Brzozowski†, Galina Jirásková‡, Baiyu Li†, and Joshua Smith†

Abstract

A language L is prefix-free if whenever words u and v are in L and u is a prefix of v, then $u = v$. Suffix-, factor-, and subword-free languages are defined similarly, where by "subword" we mean "subsequence", and a language is bifix-free if it is both prefix- and suffix-free. These languages have important applications in coding theory.

The quotient complexity of an operation on regular languages is defined as the number of left quotients of the result of the operation as a function of the numbers of left quotients of the operands. The quotient complexity of a regular language is the same as its state complexity, which is the number of states in the complete minimal deterministic finite automaton accepting the language.

The state/quotient complexity of operations in the classes of prefix- and suffix-free languages has been studied before. Here, we study the complexity of operations in the classes of bifix-, factor-, and subword-free languages. We find tight upper bounds on the quotient complexity of intersection, union, difference, symmetric difference, concatenation, star, and reversal in these three classes of languages.

Keywords: bifix-free, factor-free, finite automaton, quotient complexity, regular language, state complexity, subword-free, tight upper bound

1 Introduction

The state complexity of a regular language L is the number of states in a minimal deterministic finite automaton (dfa) accepting L [41]. This complexity is the same as the quotient complexity [5] of L, which is the number of distinct left quotients of L. We prefer quotient complexity since it is more closely related to properties of languages. The quotient complexity of an operation in a class $\mathcal{C}$ of regular languages is the maximal quotient

*This work was supported by the Natural Sciences and Engineering Research Council of Canada under grant no. OGP0000871, by the Slovak Research and Development Agency under contract APVV-0035-10 "Algorithms, Automata, and Discrete Data Structures", and by VEGA grant 2/0183/11.

†David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada N2L 3G1. Baiyu Li's present address: Optumsoft, Inc., 275 Middlefield Rd, Suite 210, Menlo Park, CA 94025, USA. Joshua Smith's present address: Spielo International Canada ULC 328 Urquhart Ave. Moncton NB, Canada E1H 2R6. E-mail: {brzozo,b5li,}@uwaterloo.ca, belowtwenty@hotmail.com

‡Mathematical Institute, Slovak Academy of Sciences, Grešákova 6, 040 01 Košice, Slovakia. E-mail: jiraskov@saske.sk

DOI: 10.14232/actacyb.21.4.2014.1

complexity of the language resulting from the operation, taken as a function of the quotient complexities of the operands in class $\mathcal{C}$. For more information on state and quotient complexity see [5, 6, 41].

One of the first results concerning the state complexity of an operation is the 1966 theorem by Mirkin [33], who showed that the bound 2^n for the reversal of an n-state dfa can be attained. In 1970 Maslov [32] stated without proof the bounds on the complexities of union, concatenation, star, and several other operations in the class of regular languages, and gave languages meeting these bounds. In 1994 these operations, along with intersection, reversal, and left and right quotients, were studied in detail by Yu, Zhuang and K. Salomaa [42].

State complexity of operations has also been studied in several proper subclasses of regular languages. Surprisingly, in the class of star-free languages studied by Brzozowski and Liu [10], the operations union, intersection, difference, symmetric difference, concatenation and star meet the bounds for arbitrary regular languages; in the case of reversal, the bound 2^n cannot be reached [11], but $2^n - 1$ is attainable. In general, however, the bounds are quite different in different classes. In addition to the star-free class, the following classes have been considered: unary languages (Yu, Zhuang and K. Salomaa [42], Pighizzini and Shallit [35]); finite languages (Câmpeanu, Culik, K. Salomaa and Yu [12], Yu [41], Han and K. Salomaa [16]); cofinite languages (Bassino, Giambruno and Nicaud [2]); right-, left-, two-sided and all-sided ideals (Brzozowski, Jirásková and Li [7]); prefix-, suffix-, factor- and subword-closed languages (Brzozowski, Jirásková and Zou [9]); union-free languages (Jirásková and Masopust [22], Jirásková and Nagy [24]); non-returning languages (Eom, Han and Jirásková [14]); reversal in $\mathcal{R}$-trivial and $\mathcal{J}$-trivial languages (Jirásková and Masopust [23]); and operations on prefix- and suffix-free languages discussed below in more detail.

Languages that are prefix-, suffix-, bifix-, factor- (also called infix-), and subword-free will be called *xfix-free*. Xfix-free languages (with the exception of $\{\varepsilon\}$, where ε is the empty word) are codes, which constitute an important class of languages and have applications in such areas as cryptography, data compression, and information transmission. They have been studied extensively; see, for example, [3, 27]. In particular, *prefix* and *suffix codes* [3] are prefix-free and suffix-free languages, respectively, *infix codes* [36, 37] are factor-free, and *hypercodes* [36, 37] are subword-free, where by subword we mean subsequence. Moreover, xfix-free languages are special cases of convex[1] languages [1, 38]. We are interested only in regular xfix-free languages.

The state complexities of intersection, union, concatenation, star, and reversal for prefix-free languages were first studied by Han, K. Salomaa and Wood [18]. These results have been strengthened by Jirásková and Krausová in [21] who provided witnesses over smaller alphabets. The same operations for suffix-free languages were studied by Han and K. Salomaa [17]. The upper bounds for suffix-free languages from [17] have been shown to be tight in the binary case for union and intersection by Jirásková and Olejár [25], and for star by Cmorik [13]. On the other hand, as shown in [13], the upper bound for reversal of suffix-free languages cannot be met in the binary case. In [13, 20, 21, 30], some

[1] A language is prefix-convex if it satisfies the condition that, if a word w and its prefix u are in the language, then so is every prefix of w that has u as a prefix. In a similar way, we define suffix-, bifix-, factor-, and subword-convex languages.

other operations on prefix- and suffix-free languages, such as difference, symmetric difference, left quotient, and cyclic shift have been studied, and tight bounds with witnesses over optimal alphabets have been found.

In this paper, we study bifix-, factor- and subword-free languages. In particular, we obtain tight upper bounds on the complexities of intersection, union, difference, symmetric difference, star, product (concatenation), and reversal in these three classes of xfix-free languages.

A much shorter version of this paper appeared in [8]. In the present paper we have added several new results on binary bifix- and factor-free languages.

2 Preliminaries

It is assumed that the reader is familiar with finite automata and regular languages as treated in [34, 40], for example. If Σ is a finite non-empty *alphabet*, then Σ^* is the set of all words over this alphabet, with ε as the *empty word*. For $w \in \Sigma^*$, let $|w|$ be the length of w. A *language* is any subset of Σ^*.

The following set operations are defined on languages: *complement* ($\overline{L} = \Sigma^* \setminus L$), *union* ($K \cup L$), *intersection* ($K \cap L$), *difference* ($K \setminus L$), and *symmetric difference* ($K \oplus L$). All four of these Boolean operation with two arguments are denoted by $K \circ L$.

We also define the *product*, usually called *concatenation* or *catenation*, ($KL = \{w \in \Sigma^* \mid w = uv, u \in K, v \in L\}$), (Kleene) *star* ($L^* = \bigcup_{i \geq 0} L^i$ with $L^0 = \{\varepsilon\}$), and *positive closure* ($L^+ = \bigcup_{i \geq 1} L^i$).

The *reverse* w^R of a word $w \in \Sigma^*$ is defined inductively as follows: $\varepsilon^R = \varepsilon$, and $(wa)^R = aw^R$ for every symbol a in Σ and every word w in Σ^*. The *reverse* of a language L is denoted by L^R and is defined as $L^R = \{w^R \mid w \in L\}$.

Regular languages over Σ are languages that can be obtained from the *set of basic languages* $\{\varnothing, \{\varepsilon\}\} \cup \{\{a\} \mid a \in \Sigma\}$, using a finite number of operations of union, product, and star. We use regular expressions to represent languages. If E is a regular expression, then $\mathcal{L}(E)$ is the language denoted by that expression. For example, the regular expression $E = (\varepsilon \cup a)^* b$ denotes language $L = \mathcal{L}(E) = (\{\varepsilon\} \cup \{a\})^* \{b\}$. We usually do not distinguish notationally between regular languages and regular expressions.

The *ε-function* L^ε of a regular language L is $L^\varepsilon = \varnothing$ if $\varepsilon \notin L$, and $L^\varepsilon = \{\varepsilon\}$ if $\varepsilon \in L$. We usually write the language $\{\varepsilon\}$ as ε. Then the ε-function can be computed inductively as follows:

$$a^\varepsilon = \begin{cases} \varnothing, & \text{if } a = \varnothing, \text{ or } a \in \Sigma; \\ \varepsilon, & \text{if } a = \varepsilon. \end{cases} \tag{1}$$

$$(\overline{L})^\varepsilon = \begin{cases} \varnothing, & \text{if } L^\varepsilon = \varepsilon; \\ \varepsilon, & \text{if } L^\varepsilon = \varnothing. \end{cases} \tag{2}$$

$$(K \cup L)^\varepsilon = K^\varepsilon \cup L^\varepsilon, \quad (KL)^\varepsilon = K^\varepsilon \cap L^\varepsilon, \quad (L^*)^\varepsilon = \varepsilon. \tag{3}$$

The *quotient* [4] of a language L by a word w is defined as $L_w = \{x \in \Sigma^* \mid wx \in L\}$. Quotients of regular languages [4, 5] can be computed as follows. The *quotient by a letter*

a in Σ is computed by induction:

$$b_a = \begin{cases} \varnothing, & \text{if } b \in \{\varnothing, \varepsilon\}, \text{ or } b \in \Sigma \text{ and } b \neq a; \\ \varepsilon, & \text{if } b = a. \end{cases} \tag{4}$$

$$(\overline{L})_a = \overline{L_a}, \quad (K \cup L)_a = K_a \cup L_a, \quad (KL)_a = K_a L \cup K^\varepsilon L_a, \quad (L^*)_a = L_a L^*. \tag{5}$$

The quotient by a word w in Σ^* is computed by induction on the length of w:

$$L_\varepsilon = L, \quad L_{wa} = (L_w)_a. \tag{6}$$

By convention, L_w^ε always means $(L_w)^\varepsilon$.

A *deterministic finite automaton* (dfa) is a quintuple $\mathcal{D} = (Q, \Sigma, \delta, q_0, F)$, where Q is a finite non-empty set of *states*, Σ is a finite *alphabet*, $\delta\colon Q \times \Sigma \to Q$ is the *transition function*, q_0 is the *initial state*, and $F \subseteq Q$ is the set of *final* states. As usual, the transition function is extended to $Q \times \Sigma^*$. Dfa $\mathcal{D}$ accepts a word w in Σ^* if $\delta(q_0, w) \in F$. The set of all words accepted by $\mathcal{D}$ is $L(\mathcal{D})$, the language accepted by $\mathcal{D}$. By the *language of a state* q of $\mathcal{D}$ we mean the language L_q accepted by the automaton $(Q, \Sigma, \delta, q, F)$. Two states of $\mathcal{D}$ are *equivalent* if their languages are equal. A state is *empty* if its language is empty.

A quotient L_w is *final* if $\varepsilon \in L_w$; otherwise it is *non-final*. The *quotient automaton* of a regular language L is the automaton in which the quotients of the language are states; it is formally defined as the dfa $\mathcal{D} = (Q, \Sigma, \delta, q_0, F)$, where $Q = \{L_w \mid w \in \Sigma^*\}$, $\delta(L_w, a) = L_{wa}$, $q_0 = L_\varepsilon$, $F = \{L_w \mid \varepsilon \in L_w\}$. This is a minimal dfa accepting L. The number of distinct quotients of a language is called its *quotient complexity* and is denoted by $\kappa(L)$. Hence the quotient complexity of L is equal to the state complexity of L, and we call it simply *complexity*. Whenever convenient, we derive upper bounds on the quotient complexity of operations on xfix-free languages following the approach of [5].

A *nondeterministic finite automaton* (nfa) is a quintuple $\mathcal{N} = (Q, \Sigma, \delta, I, F)$, where Q, Σ, and F are as in a dfa, $\delta\colon Q \times \Sigma \to 2^Q$ is the nondeterministic transition function, and I is the set of initial states. We extend the transition function to a function $\delta\colon 2^Q \times \Sigma^* \to 2^Q$ as usual. The language accepted by $\mathcal{N}$ is $L(\mathcal{N}) = \{w \in \Sigma^* \mid \delta(I, w) \cap F \neq \varnothing\}$. Two nfas are *equivalent* if their languages are equal.

A *reverse* of a dfa $\mathcal{D} = (Q, \Sigma, \delta, q_0, F)$ is an nfa $\mathcal{D}^R = (Q, \Sigma, \delta^R, F, \{q_0\})$, where $\delta^R(q, a) = \{p \in Q \mid \delta(p, a) = q\}$. The nfa $\mathcal{D}^R$ accepts the language $(L(\mathcal{D}))^R$.

Every nondeterministic finite automaton $\mathcal{N} = (Q, \Sigma, \delta, I, F)$ can be converted to an equivalent dfa $\mathcal{D} = (2^Q, \Sigma, \delta', I, F')$, where $F' = \{R \in 2^Q \mid R \cap F \neq \varnothing\}$ and $\delta'(R, a) = \cup_{r \in R} \delta(r, a)$ for each R in 2^Q and each a in Σ. We call this dfa $\mathcal{D}$ the *subset automaton* of nfa $\mathcal{N}$. The subset automaton need not be minimal since some of its states may be unreachable or equivalent.

3 Xfix-Free Languages

If $u, v, w, x \in \Sigma^*$ and $w = uxv$, then u is a *prefix* of w, x is a *factor* of w, and v is a *suffix* of w. Both u and v are also factors of w. If $w = u_0 v_1 u_1 \cdots v_n u_n$, where $u_i, v_i \in \Sigma^*$, then $v = v_1 v_2 \cdots v_n$ is a *subword* of w. Every factor of w is also a subword of w.

A language L is *prefix-free* (respectively, *suffix-*, *factor-*, or *subword-free*) if, whenever words u and v are in L and u is a prefix (respectively, suffix, factor, or subword) of v, then $u = v$. Additionally, L is *bifix-free* if it is both prefix- and suffix-free. All subword-free languages are factor-free, and all factor-free languages are bifix-free.

If ε is a quotient of L, then L also has the empty quotient, since $\varepsilon_a = \varnothing$, for all a in Σ. We say that a quotient L_w is *uniquely reachable* if $L_w = L_x$ implies that $w = x$. We now restate two results from [17, 18] in our terminology.

Remark 1. A non-empty language is prefix-free if and only if it has exactly one final quotient and that quotient is ε.

Remark 2. If L is a non-empty suffix-free language, then it has the empty quotient and $L_\varepsilon = L$ is uniquely reachable.

Let L be any language. If $(L_u)_x = L_v$ for some words u, v and a non-empty word x, then L_v is *positively reachable* from L_u, and we denote this by $L_u \to L_v$. The relation $\to$ is transitive. The next remark uses this relation to characterize finite languages.

Remark 3. If L is any language with $\{L_1, L_2, \ldots, L_n\}$ as its set of quotients, and u and v are words in Σ^*, then the following are equivalent:

1. L is finite.
2. $L_u \to L_v$ and $L_v \to L_u$ if and only if $L_u = L_v = \varnothing$.
3. There exists a total order $L = L_1 \prec L_2 \prec \cdots \prec L_{n-1} \prec L_n = \varnothing$, on the set of quotients such that for any $x \in \Sigma^+$, $(L_i)_x = L_j$ implies $L_i \prec L_j$ or $L_i = L_j = L_n$.

Note that every subword-free language is finite [15]. The next lemma will be used later to prove that upper bounds on the quotient complexity of some operations on subword-free languages cannot be reached if the alphabet of the language does not have sufficiently many letters.

Remark 4. Let L be a finite language with $\kappa(L)$, where $n \geq 4$. Let the distinct quotients $L = L_\varepsilon = L_1, L_2, \ldots, L_{n-2}, L_{n-1} = \varepsilon, L_n = \varnothing$ of L be ordered as in Remark 3. If $L_w = L_2$ for some word w, then $|w| = 1$.

Finally, we describe a simple method of constructing xfix-free languages.

Proposition 1. *Let $L \subseteq \Sigma^*$ be any language, and let $a \notin \Sigma$. Then (1) aL is suffix-free, (2) La is prefix-free, (3) aLa is factor-free.*

4 Boolean Operations

For prefix-free languages, it was shown in [18, 21] that the tight bounds for union, intersection, difference, and symmetric difference are $mn-2$, $mn-2(m+n-3)$, $mn-(m+2n-4)$, and $mn-2$, respectively. For union and intersection of suffix-free languages, the tight bounds are $mn-(m+n-2)$ and $mn-2(m+n-3)$, respectively [17]. The bounds

for difference and symmetric difference are $mn-(m+2n-4)$ and $mn-(m+n-2)$, respectively [25], and the bounds for all four Boolean operations are met by binary suffix-free languages [13]. The next theorem provides results for Boolean operations on bifix- and factor-free languages.

Theorem 1 (Boolean Operations: Bifix- and Factor-Free Languages). *Let K and L be bifix-free languages over an alphabet Σ with $\kappa(K)=m$ and $\kappa(L)$, where $m,n\geq 4$. Then*

1. $\kappa(K\cap L)\leq mn-3(m+n-4)$;
2. $\kappa(K\setminus L)\leq mn-(2m+3n-9)$;
3. $\kappa(K\cup L),\kappa(K\oplus L)\leq mn-(m+n)$.

The bounds for intersection and difference can be met by binary factor-free languages, and the bound for union and symmetric difference can be met by ternary factor-free languages.

Proof. We first derive the upper bound for bifix-free languages; this bound applies also to factor-free languages. Since $(K\circ L)_w=K_w\circ L_w$, the bound for regular languages is mn; however, not all pairs (K_i,L_j) of quotients of K and L can occur if the languages are xfix-free.

If K and L are bifix-free, by unique reachability we get a reduction of $m-1+n-1=m+n-2$ from the general bound mn, because pairs of the form (K_ε,L_w) and (K_w,L_ε) can occur only if $w=\varepsilon$.

Moreover, both languages K and L have ε and $\varnothing$ as quotients. For intersection, we have $\varnothing\cap L_w=K_w\cap\varnothing=\varnothing$, and this results in another reduction of $m-2+n-2$ quotients. Also, the quotients $\varepsilon\cap L_w$ and $K_w\cap\varepsilon$ are either empty or equal to ε; this gives an additional reduction of $m-3+n-3$ states. Altogether, we get the upper bound.

For difference, we eliminate $m+n-2$ quotients by unique reachability, $n-2$ quotients by the fact that $\varnothing\setminus L_w=\varnothing$ (keeping only one representative $\varnothing\setminus\varnothing$), $m-2$ quotients by the fact that $K_w\setminus\varnothing=K_w\setminus\varepsilon$ (keeping $K_w\setminus\varnothing$ as a representative), and $n-3$ more quotients by the rule $\varepsilon\setminus L_w=\varepsilon$, for a total reduction of $2m+3n-9$. For union we have the unique reachability reduction of $m+n-2$, and a further reduction of 2 by the rule $\varepsilon\cup\varepsilon=\varepsilon\cup\varnothing=\varnothing\cup\varepsilon=\varepsilon$. For symmetric difference we have the unique reachability reduction of $m+n-2$, and a further reduction of 2 in view of the fact that $\varepsilon\oplus\varepsilon=\varnothing\oplus\varnothing=\varnothing$ and $\varepsilon\oplus\varnothing=\varnothing\oplus\varepsilon=\varepsilon$.

For the tightness of the bounds for intersection and difference, let K and L be the binary factor-free languages accepted by the quotient automata of Figure 1, where missing transitions in the automaton accepting K (L) all go to the empty state m (n). In the corresponding cross-product automaton of Figure 2, no states in row 1 or column 1 are reachable, except for $(1,1)$. Also, states $(m-1,2)$ and $(m,2)$ are unreachable, as are the states in column $n-1$, except $(3,n-1)$, $(m-1,n-1)$, and $(m,n-1)$. The remaining states are all reachable.

For intersection, the only final state is $(m-1,n-1)$, and all the other states in the last two rows and columns are empty. We will prove that states $(1,1)$, (i,j) with $2\leq i\leq m-2$

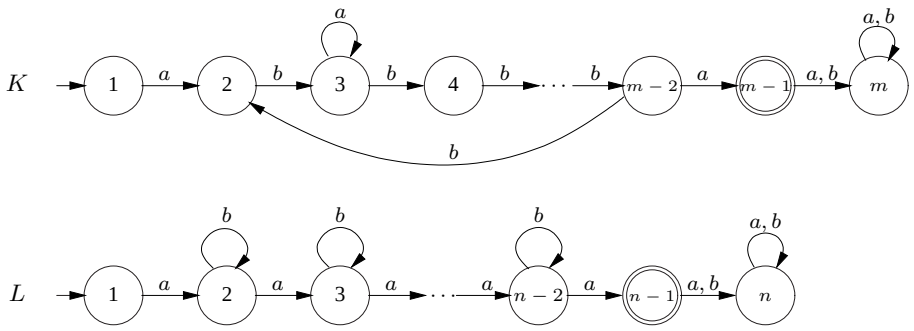

Figure 1: Binary factor-free witnesses for intersection and difference.

and $2 \leq j \leq n-2$, $(m-1, n-1)$, and (m, n) (which represents all the empty states) are all distinguishable. Then it follows that $\kappa(K \cap L) \geq (m-3)(n-3)+3 = mn-3(m+n-4)$.

State (m, n) is the only empty state in our set. We show that for each other non-final state (i, j), there exists a word w_{ij} that is accepted only from state (i, j). We have $w_{m-2,n-2} = a$ because word a is accepted only from state $(m-2, n-2)$. Since only one transition on letter b goes to state $(m-2, n-2)$, and it goes from state $(m-3, n-2)$, the word ba is accepted only from state $(m-3, n-2)$. Therefore $w_{m-3,n-2} = ba = bw_{m-2,n-2}$. For similar reasons we have

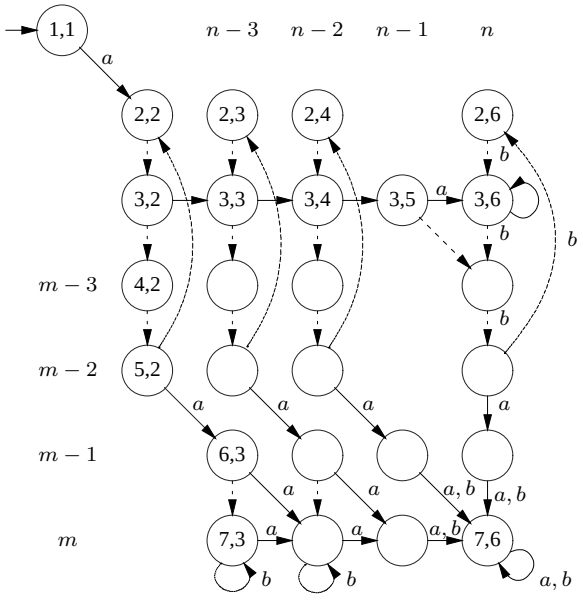

Figure 2: Cross-product automaton for $m = 6, n = 7$. Missing transitions go to $(7, 6)$.

$w_{i,n-2} = bw_{i+1,n-2}$ for $i = 2, 3, \ldots, m-3$,
$w_{3j} = aw_{3,j+1}$ for $j = 2, 3, \ldots, n-3$,
$w_{2j} = bw_{3j}$ for $j = 2, 3, \ldots, n-3$,
$w_{m-2,j} = bw_{2j}$ for $j = 2, 3, \ldots, n-3$,
$w_{ij} = bw_{i+1,j}$ for $i = 4, 5, \ldots, m-3$ and $j = 2, 3, \ldots, n-3$,
$w_{11} = aw_{22}$,

which proves that $mn - 3(m+n-4)$ states are pairwise distinguishable.

In the case of difference, all the states in row m, as well as state $(m-1, n-1)$ are empty. All the other states in row $m-1$ accept ε, and so are equivalent. For each i with $2 \leq i \leq m-2$, states $(i, n-1)$ and (i, n) are equivalent. Among the other reachable states consider two distinct states p and q. If they are in different rows, then by a word in b^* we can send p to a state p' in row 3, and q to a state q' that is not in row 3. Now by a^n, state q' goes to the empty state, while p' goes to state $(3, n)$ that is not empty. Two distinct states in the same row go by a word in b^* to row 3. Then, by a word in a^*, the first goes to state $(3, n-2)$ while the second to $(3, n)$, and now $b^{m-2-3}a$ distinguishes them. In summary, $\kappa(K \setminus L) \geq (m-3)(n-3) + m - 3 + 3 = mn - (2m + 3n - 9)$.

To prove the tightness of the bounds for union and symmetric difference, consider the languages $K = a(c^*(a \cup b))^{m-3}$, $L = a(b^*(a \cup c))^{n-3}$; see Figure 3, where missing transitions in the automaton accepting K (L) all go to the empty state m (n). If $w \in K$, then $w = av$ for some word v containing $m-3$ occurences of symbols from $\{a, b\}$ and ending in a or b. Thus no proper factor of w is in K, and so K is factor-free. A similar proof applies to L.

In the cross-product automaton of Figure 4 for the Boolean operations on languages K and L, all the states are reached from the initial state $(1, 1)$ by a word in $ab^*c^* \cup ac^*b^*$, except for state $(m-1, n-1)$ which is reached from state $(m-2, n-2)$ by a.

For union, all the states in row $m-1$ and in column $n-1$ are final, and moreover, the three states $(m, n-1)$, $(m-1, n-1)$, and $(m-1, n)$ are equivalent. The word ab^{m-3} is accepted only from $(1, 1)$. Consider two distinct non-final states (i, j) and (k, ℓ). If $i < k$, then $c^n b^{m-1-i}$ is accepted from (i, j) but not from (k, ℓ). If $j < \ell$, then $b^m c^{n-1-j}$ is accepted from (i, j) but not from (k, ℓ). Now consider two distinct final states different

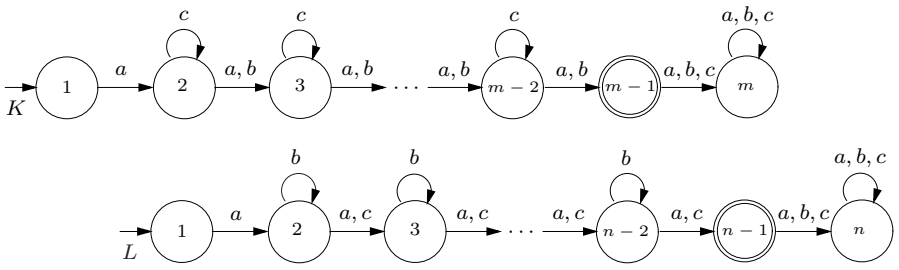

Figure 3: Ternary factor-free languages witnesses for union and symmetric difference.

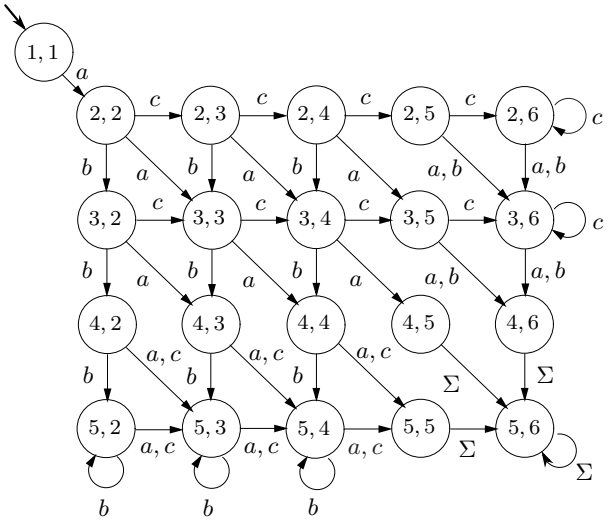

Figure 4: Cross-product automaton for Boolean operations on languages from Figure 3.

from $(m, n-1)$ and $(m-1, n)$. By c, the two states either go to two states one of which is final and the other non-final, or to two distinct non-final, and hence distinguishable, states. This proves distinguishability of $mn - (m+n)$ states.

The proof for symmetric difference is the same as for union, except that now state $(m-1, n-1)$ is empty and states $(m, n-1)$ and $(m-1, n)$ are equivalent. □

The next proposition gives lower bounds for union and symmetric difference of binary bifix-free languages.

Proposition 2 (Union, Symmetric Difference: Binary Bifix-Free Languages; Lower Bound). *Let $m, n \geq 6$. There exist binary bifix-free languages K and L with $\kappa(K) = m$ and $\kappa(L) = n$ such that $\kappa(K \cup L), \kappa(K \oplus L) \geq mn - (m+n) - 2$.*

Proof. Consider the binary languages

$$\begin{aligned} K &= a((ba^*)^{m-5}b \cup a)(b((ba^*)^{m-5}b \cup a))^*a, \\ L &= a(a \cup b)^{n-4}(b(a \cup b)^{n-4})^*a. \end{aligned}$$

Quotient automata for $m = 7$ and $n = 6$ are shown in Figure 5. Since both languages have ε as the only final quotient, they are prefix-free. Since the reverse automata are deterministic, the reversed languages also have ε as the only final quotient, and so are prefix-free. Thus both languages are bifix-free.

The cross-product automaton is shown in Figure 6. States in row 1 and column 1 are unreachable, with the exception of the initial state (1,1). Also, states $(2, n-1)$ and

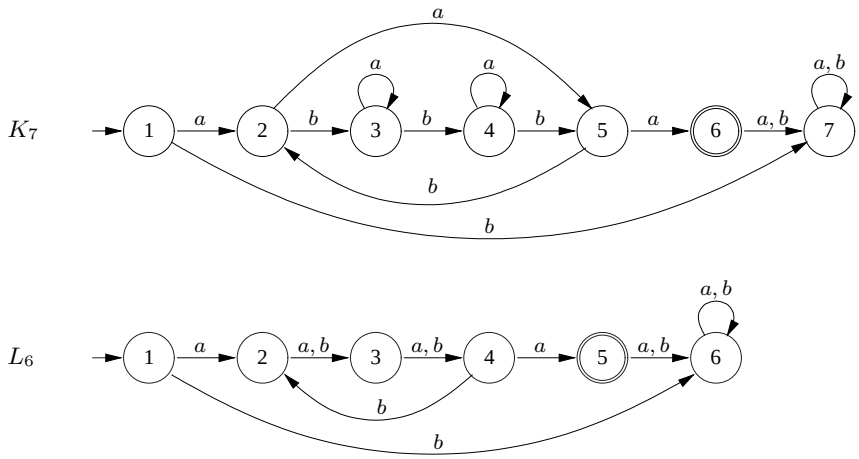

Figure 5: Binary bifix-free languages meeting the bound $mn-(m+n)-2$ for union and symmetric difference.

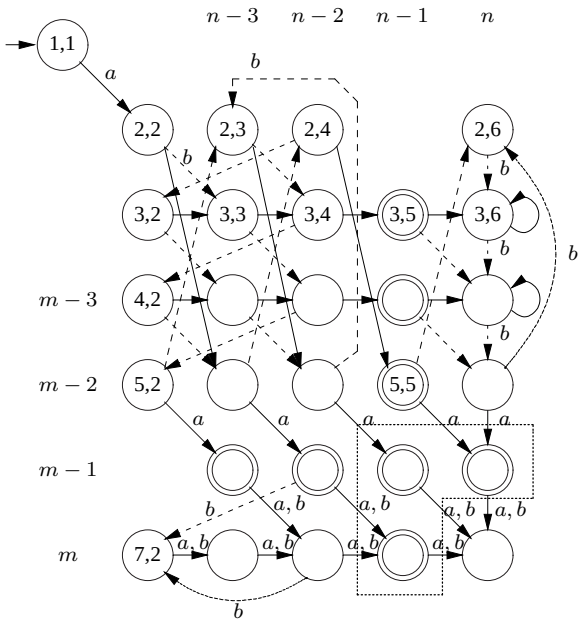

Figure 6: Cross-product automaton for automata from Figure 5, where dashed-transitions are on input b, and missing transitions go to state (7,6).

$(m-1,2)$ are unreachable. The initial state $(1,1)$ goes to state $(2,2)$ by a and then to state $(3,3)$ by b. From $(3,3)$, all the other states in row 3, except for $(3,2)$ are reached by a-transitions. Next, state $(3,n-2)$ goes to state $(4,2)$ by b, and then to $(4,j)$ by a^{j-2} $(3 \leq j \leq n)$. In this way, all the states in rows $4,5,\ldots,m-3$ can be reached. State $(m-3,n-2)$ goes to state $(m-2,2)$ by b, and states $(m-2,j)$ with $j \geq 3$, except for state $(m-2,n-1)$ that is reached from $(2,n-2)$ by a, are reached from states $(m-3,j-1)$ by b. States $(2,j)$ with $j \geq 3$, except for $(2,n-1)$, are reached from $(m-2,j-1)$ by b. State $(2,n-2)$ goes to $(3,2)$ by b. From states in row $m-2$ all reachable states in row $m-1$ are reached by a. State $(m,2)$ is reached by b from $(m-1,n-2)$; from here, all the other states is row m are reached by words in a^*.

For union, the three final states $(m-1,n-1),(m-1,n)$ and $(m,n-1)$ are equivalent. Consider the other reachable states. First, let $p=(i,j)$ and $q=(k,\ell)$ be two non-final states with $i<k$. We can use b-transitions to get p into a state p' in row 3, and q into a state q' in a row i with $i \neq 3$. By a^n, state p' goes to $(3,n)$, while q' goes to (i,n). Now $b^{m-2-3}a$ is accepted from $(3,n)$ but not from (i,n). Next, let p and q be two distinct non-final states in the same row. If they are in the last row, then a word in a^* distinguishes them. Otherwise, we can get them into states $(3,j)$ and $(3,\ell)$ with $j<\ell$, using b-transitions. Now $(3,j)$ accepts a^{n-1-j} while $(3,\ell)$ goes to the non-final state $(3,n)$. Finally, consider two distinct final states different from $(m-1,n),(m,n-1)$. By b, they go to two distinct non-final, and so distinguishable, states. The proof for symmetric difference is similar, except that now state $(m-1,n-1)$ is empty. □

We now show that the upper bound for union and symmetric difference of binary bifix-free languages is the same as the lower bound in the proposition above.

Proposition 3 (Union, Symmetric Difference: Binary Bifix-Free Languages; Upper Bound). *Let $m,n \geq 4$ and let K and L be binary bifix-free languages with $\kappa(K)=m$ and $\kappa(L)$. Then $\kappa(K \cup L), \kappa(K \oplus L) \leq mn-(m+n)-2$.*

Proof. Let K be a bifix-free language accepted by the quotient automaton $\mathcal{A}$ over $\{a,b\}$ with states $1,2,\ldots,m$, where 1 is the initial state, $m-1$ is the only final state and it accepts only ε, and m is the empty state. Let L be a similar language accepted by $\mathcal{B}$ with states $1,2,\ldots,n$, initial state 1, final state $n-1$ accepting ε, and empty state n.

Construct the corresponding cross-product automaton with states (i,j), where i is a state of $\mathcal{A}$ and j is a state of $\mathcal{B}$. In this cross-product automaton, we cannot go from rows $m-1$ and m to any state (i,j) with $i<m-1$, and similarly, we cannot go from columns $n-1$ and n to and state (i,j) with $j<n-1$.

If state 1 of $\mathcal{A}$ goes by both inputs a and b to a state in $\{m-1,m\}$, then no row i with $i<m-1$ can be reached. Therefore, if the bound is to be met, at least one input, say a, takes state 1 to a state i with $i<m-1$. Suppose also that b takes 1 to a state in $\{m-1,m\}$. A similar condition applies to L. Suppose that input b takes state 1 of $\mathcal{B}$ to a state j with $j<n-1$, and a, to a state in $\{n-1,n\}$. Then no state (i,j) with $i<m-1$ and $j<n-1$ can be reached. It follows that, without loss of generality, each automaton must take its initial state by a to a state that is neither final nor empty; for convenience, let this state be 2 in both automata. Then no other transition by a may go to state 2 in the two automata, otherwise they would not be suffix-free.

It follows that in the cross-product automaton, all the states in row 2 and column 2, except for $(2,2)$, must be reached from some states by input b. Thus, if all the states are reachable, there must be an incoming transition by b to each state i with $i \geq 2$ in $\mathcal{A}$ and j with $j \geq 2$ in $\mathcal{B}$. In particular, if state $(m-1,2)$ or $(2,n-1)$ is reachable, then some state, say p_1 (respectively q_1) different from $m-1$ (respectively $n-1$) must go to state $m-1$ (respectively $n-1$) in $\mathcal{A}$ (respectively $\mathcal{B}$). Now since p_1 goes to $m-1$ by b, it cannot go anywhere else by b. Thus there must be some other state p_2 not in $\{p_1, m-1, m\}$ that goes to p_1 by b. Then there must be a state p_3 not in $\{p_2, p_1, m-1, m\}$ that goes to p_2 by b, and so on. Eventually, we have $p_{m-3} \xrightarrow{b} p_{m-4} \xrightarrow{b} \cdots \xrightarrow{b} p_3 \xrightarrow{b} p_2 \xrightarrow{b} p_1 \xrightarrow{b} m-1 \xrightarrow{b} m$, where all the states are pairwise distinct, and no state, except possibly state 1, goes by b to state p_{m-3}.

First assume state 1 goes to state p_{m-3} by b. If $p_{m-3} = 2$, then 1 goes to 2 by a and by b. This means that there is no other transition to state 2, and so row 2 is not reachable in the cross-product automaton. If $p_{m-3} > 2$ and 1 goes to p_{m-3} by b, then no other state goes to p_{m-3} by b because of suffix-freeness, and so row p_{m-3} may only be reached by a's. However, in such a case $(p_{m-3}, 2)$ is unreachable, since it is in row p_{m-3} that can be reached only by a's and at the same time in column 2 that can be reached only by b's.

Now assume that there is no transition by b going to state p_{m-3}. If $p_{m-3} \geq 3$, then $(p_{m-3}, 2)$ is unreachable. If $p_{m-3} = 2$, then the whole row 2, except for $(2,2)$ is unreachable. The same considerations hold for automaton $\mathcal{B}$. This gives the desired upper bound $mn-(m+n)-2$. □

As a corollary of the two propositions above, we get the tight bound on the complexity of union and symmetric difference of binary bifix-free languages.

Theorem 2 (Union, Symmetric Difference: Binary Bifix-Free Languages). *Let K and L be binary bifix-free languages with $\kappa(K) = m$ and $\kappa(L)$, where $m, n \geq 6$. Then $\kappa(K \cup L), \kappa(K \oplus L) \leq mn-(m+n)-2$, and the bound is tight.*

In a recent paper [19] Iván has shown that $f(m,n) = mn-(m+n)-2-\lfloor \frac{\min\{m,n\}-2}{2} \rfloor$ is a lower bound on the union of binary factor-free languages, and that $f(m,n)-1$ is a lower bound for symmetric difference.

We now turn our attention to subword-free languages. The next theorem gives tight bounds for all four Boolean operations and shows that the bounds cannot be met using a fixed alphabet.

Theorem 3 (Boolean Operations: Subword-Free Languages). *Suppose that K and L are subword-free languages over an alphabet Σ with $\kappa(K) = m$ and $\kappa(L)$, where $m, n \geq 4$. Then*

1. *$\kappa(K \cup L), \kappa(K \oplus L) \leq mn-(m+n)$, and the bound is tight if $|\Sigma| \geq m+n-3$;*
2. *$\kappa(K \cap L) \leq mn-3(m+n-4)$, and the bound is tight if $|\Sigma| \geq m+n-7$;*
3. *$\kappa(K \setminus L) \leq mn-(2m+3n-9)$, and the bound is tight if $|\Sigma| \geq m+n-6$.*

Moreover, the bounds cannot be met for smaller alphabets.

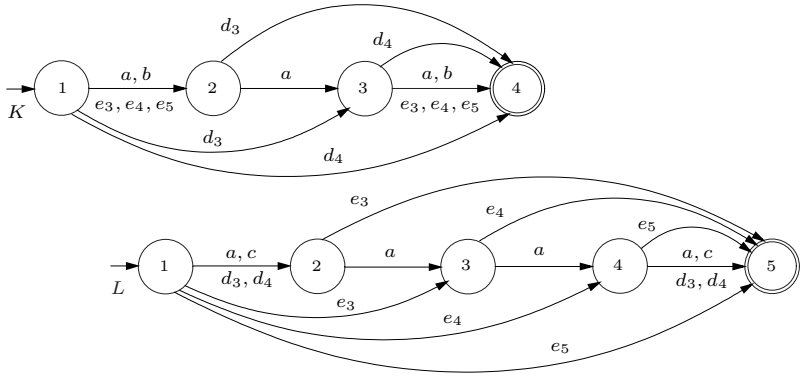

Figure 7: Subword-free witness languages for Boolean operations; $m = 5$, $n = 6$.

Proof. Since subword-free languages are bifix-free, all the upper bounds apply. To prove tightness, let $\Sigma = \{a, b, c\} \cup \{d_i \mid 3 \le i \le m-1\} \cup \{e_j \mid 3 \le j \le n-1\}$. Consider the languages K and L defined by the following quotient equations:

$$\begin{array}{lcll}
K_1 & = & (a \cup b \cup e_3 \cup \cdots \cup e_{n-1})K_2 \cup \bigcup_{i=3}^{m-1} d_i K_i, & \\
K_i & = & aK_{i+1} \cup d_{i+1}K_{m-1} & i = 2, 3, \ldots, m-3, \\
K_{m-2} & = & (a \cup b \cup d_{m-1} \cup e_3 \cup e_4 \cup \cdots \cup e_{n-1})K_{m-1}, & \\
K_{m-1} & = & \varepsilon, & \\
K_m & = & \varnothing, &
\end{array}$$

$$\begin{array}{lcll}
L_1 & = & (a \cup c \cup d_3 \cup \cdots \cup d_{m-1})L_2 \cup \bigcup_{j=3}^{n-1} e_j L_j, & \\
L_j & = & aL_{j+1} \cup e_{j+1}L_{n-1} & j = 2, 3, \ldots, n-3, \\
L_{n-2} & = & (a \cup c \cup e_{n-1} \cup d_3 \cup d_4 \cup \cdots \cup d_{m-1})L_{n-1}, & \\
L_{n-1} & = & \varepsilon, & \\
L_n & = & \varnothing. &
\end{array}$$

The dfa's (minus empty states) for languages K and L, where $m = 5$ and $n = 6$, are shown in Figure 7. We now show that languages K and L are subword-free. For this purpose, let

$$\Gamma = \{a, b, e_3, e_4, \ldots, e_{n-1}\}, \text{ and } \Delta = \{d_3, d_4, \ldots, d_{m-1}\}.$$

Notice that no word in Γ^* of length less than $m-2$ is in K. Now let w be a word in language K. Then word w either contains no letter from Δ, or contains at most two such letters. If w contains no letter from Δ, then w is a word in Γ^* of length $m-2$, and so no its proper subword is in K. If w contains exactly one letter from Δ, then either $w = ud_i$ for some word u in Γ^* of length $i-2$, or $w = d_i v$ for some word v in Γ^* of length $m-1-i$. In both cases, no proper subword of w is in language K. Finally, if w contains two letters from Δ, then $w = d_i a^k d_{i+k+1}$ where $k \ge 0$ and $3 \le i < i+k+1 \le m-2$. No proper subword of such a word is in language K. This means that language K is subword-free. The proof for language L is similar.

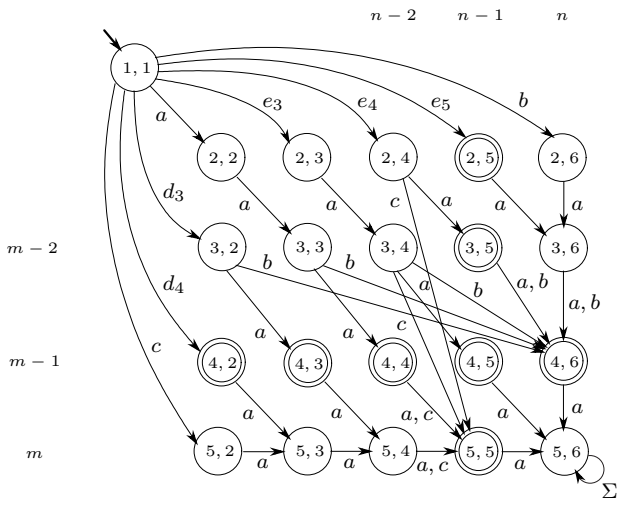

Figure 8: Reachability in the cross-product automaton for the union of languages from Figure 7 and transitions by b and c.

Figure 8 depicts the cross-product automaton of dfa's from Figure 7, where we show only the transitions necessary to prove reachability and those caused by b and c. The states in the first row and the first column, except for the initial state $(1,1)$, are unreachable. Now consider the remaining states. All the states in the second row and the second column are reached from $(1,1)$ by symbols in Σ. Each other state is reached from a state in the second row or second column by a word in a^*.

For union, all the states in row $m-1$ and in column $n-1$ are final, and the three states $(m,n-1)$, $(m-1,n-1)$, and $(m-1,n)$ accept only ε, and so are equivalent. These three states are distinguishable from all other final states, since each of the other final states accepts at least one non-empty word. Now let (i,j) and (k,ℓ) be two distinct states other than the three states accepting only word ε. First assume that $i<k$. If $i=m-1$, then state (i,j) is final while state (k,ℓ) is non-final. If $i \leq m-2$, then $a^{m-2-i}b$ is accepted from state (i,j), but not from state (k,ℓ). Symmetrically, if $j<\ell$, then either ε or $a^{n-2-j}c$ distinguishes the two states. Therefore all the $mn-(m+n)$ states are pairwise distinguishable. For symmetric difference, $(m-1,n-1)$ is empty; the rest of the proof is the same as for union.

For intersection, the only final state is $(m-1,n-1)$, and all the non-final states in the last two rows and last two columns are empty. Next, the word a is accepted only from state $(m-2,n-2)$, the word d_i $(3 \leq i \leq m-2)$ is accepted only from state $(i-1,n-2)$, while the word e_j $(3 \leq i \leq n-2)$, only from state $(m-2,j-1)$. This means that for each state (i,j), there exists a word in $a^*(a \cup d_3 \cup \cdots \cup d_{m-2} \cup e_3 \cup \cdots \cup e_{n-2})$ that is accepted only from (i,j). So we get $mn-3(m+n-4)$ pairwise distinguishable states. Notice, that here we do not use transitions by symbols b, c, d_{m-1}, e_{n-1}, and so we can

simply omit these symbols to get witness languages over an alphabet of size $m+n-7$.

For difference, all the states in row $m-1$, except for state $(m-1,n-1)$, are final and accept ε. All the states in the last row, as well as state $(m-1,n-1)$, are empty, and states $(i,n-1)$ and (i,n) with $2 \leq i \leq m-2$ are equivalent. States in different rows (up to row $m-1$) are distinguished by a word in a^*b. States in row $m-2$ are distinguished by a word in $a \cup e_3 \cup e_4 \cup \cdots \cup e_{n-2}$ because a distinguishes states $(m-2,n-2)$ and $(m-2,n-1)$, and if $2 \leq j < \ell \leq n-1$ and $j \neq n-2$, then word e_{j+1} is not accepted from $(m-2,j)$ but is accepted from $(m-2,\ell)$. Next, states $(i,n-2)$ and $(i,n-1)$ with $2 \leq i \leq m-3$ are distinguished by d_{i+1}. Finally, if two distinct states are in the same row, then there is a word in a^*, by which the two states either go to two distinct states in row $m-2$, or to two states $(i,n-2)$ and $(i,n-1)$ with $2 \leq i \leq m-3$. In both cases the resulting states are distinguishable, which proves the distinguishability of $mn-(2m+3n-9)$ states. Notice that now we do not use transitions by c, d_{m-1}, e_{n-1}, and so the bound is met for an alphabet of size $m+n-6$.

We now show that the upper bounds cannot be met using smaller alphabets. Let the quotients of K and L be $K = K_1, K_2, \ldots, K_{m-2}, K_{m-1} = \varepsilon, K_m = \varnothing$, and $L = L_\varepsilon = L_1, L_2, \ldots, L_{n-2}, L_{n-1} = \varepsilon, L_n = \varnothing$, ordered as in Remark 3. By Remark 4, all the quotients of the form $K_2 \cup L_i$ or $K_j \cup L_2$ must be reached by letters if the bound is to hold, and this is impossible if the size of the alphabet is smaller than the number of such quotients. □

5 Product and Star

The complexity of the product of prefix-free languages is $m+n-2$ [18]. For suffix-free languages, the complexity is $(m-1)2^{n-1}+1$ [17]. Since bifix-free languages are prefix-free, and the witness prefix-free languages a^{m-2} and a^{n-2} are also subword-free, we have the following result:

Theorem 4 (Product). *If K and L are bifix-free with $\kappa(K)=m$ and $\kappa(L)$, where $m,n \geq 2$, then $\kappa(KL) \leq m+n-2$. Furthermore, there are unary subword-free languages that meet this bound.*

The complexity of star is n for prefix-free languages [18], and $2^{n-2}+1$ for suffix-free languages [17]. We now extend these results to bifix-, factor-, and subword-free languages. The quotient of L^* by ε is $L^* = \varepsilon \cup LL^*$, and the following formula holds for a quotient of L^* by a non-empty word w [5]:

$$(L^*)_w = \Big(L_w \cup \bigcup_{\substack{w=uv \\ u,v \in \Sigma^+}} (L^*)^\varepsilon_u L_v\Big) L^*.$$

Theorem 5 (Star). *If L is bifix-free with $\kappa(L)$, where $n \geq 3$, then $\kappa(L^*) \leq n-1$. Furthermore, there are binary subword-free languages that meet this bound.*

Proof. Assume that L is bifix-free. Then it is prefix-free, has only one final quotient, namely ε, and has the empty quotient, by Remark 1. Moreover, since L is suffix-free, the quotient L is uniquely reachable by ε, by Remark 2.

Let L_w be a non-empty quotient of L by a non-empty word w. Let us show that $(L^*)^{\varepsilon}_u = \varnothing$ for every proper non-empty prefix u of w. Assume for contradiction that $\varepsilon \in (L^*)_u$, where $w = uv$ for some non-empty words u and v. Then $u \in L^*$, and so there exist words x in L and y in L^* such that $u = xy$. This gives $L_w = L_{xyv} = \varepsilon_{yv} = \varnothing$ because $x \in L$ implies $L_x = \varepsilon$. This is a contradiction, and so we must have $(L^*)^{\varepsilon}_u = \varnothing$. Hence, if L_w is non-empty, then $(L^*)_w = L_w L^*$, by the equation above. Now if L_w is final, then $L_w = \varepsilon$, and so $(L^*)_w = L^* = (L^*)_\varepsilon$. There are $n-2$ choices for non-final and non-empty quotients L_w. But, for a non-empty word w, we have $L_w \neq L$ since L is uniquely reachable by ε. This reduces the number of choices to $n-3$ since $n \geq 3$.

Now consider $L_w = \varnothing$ for a non-empty word w. Let u be the longest proper non-empty prefix of w such that $(L^*)^{\varepsilon}_u = \varepsilon$. If no such u exists, then $(L^*)_w = \varnothing$. Otherwise, let us show that for every proper non-empty prefix u' of u, we must have $(L^*)^{\varepsilon}_{u'} = \varnothing$. Assume for a contradiction that $(L^*)^{\varepsilon}_{u'} \neq \varnothing$. Then $u' \in L^*$ and also $u \in L^*$. So there exist $x, x' \in L$ and $y, y' \in L^*$ such that $u = xy$ and $u' = x'y'$. Since u' is a proper prefix of u, one of x and x' is a prefix of the other. If $x \neq x'$, then L is not prefix-free, which is a contradiction. If $x = x'$, then $y \neq y'$ and y' is a proper prefix of y. By an induction on the length of y' we can derive a contradiction that L is not prefix-free. So $(L^*)_w = (L^*)^{\varepsilon}_u L_v L^* = L_v L^*$, which has already been counted.

In total, there are at most $n-1$ quotients of L^*. The subword-free language a^{n-2} over $\{a, b\}$ meets the bound since the language $(a^{n-2})^*$ has $n-2$ quotients of the form $a^{n-2-i}(a^{n-2})^*$ for $i = 1, 2, \ldots, n-2$, and it has the empty quotient. □

6 Reversal

The last operation we consider is reversal. In [17, 18] it was shown that the complexity of reversal is $2^{n-2}+1$ for suffix-free or prefix-free languages. We show that this bound can be reduced for bifix-free languages. We use the standard method of reversing the quotient dfa $\mathcal{D}$ of L to obtain an nfa $\mathcal{D}^R$ for the language L^R, and then we apply the subset construction to nfa $\mathcal{D}^R$ to get a dfa for L^R.

Theorem 6 (Reversal: Bifix- and Factor-Free Languages). *If L is a bifix-free language with $\kappa(L)$, where $n \geq 3$, then $\kappa(L^R) \leq 2^{n-3}+2$. Moreover, there exist ternary factor-free languages that meet this bound.*

Proof. If L is bifix-free, then so is L^R. Since L is prefix-free, it has exactly one final quotient, ε, and also has the empty quotient.

Consider the quotient automaton $\mathcal{D}$ for L, and remove the empty quotient and all the transitions to the empty quotient. Reverse this incomplete dfa to get an $(n-1)$-state nfa $\mathcal{D}^R$ for L^R. Consider the subset automaton of the nfa $\mathcal{D}^R$. The initial state of the subset automaton is the singleton set $\{f\}$, where f is the quotient ε in the quotient automaton $\mathcal{D}$. No other subset containing state f is reachable in the subset automaton since no transition goes to state f in nfa $\mathcal{D}^R$. This gives at most $2^{n-2}+1$ reachable states. However, language L^R is prefix-free, and so all the final states of the subset automaton accept only the empty word, and can be merged into one state. Hence $\kappa(L^R) \leq 2^{n-3}+2$.

If $n = 3$ or $n = 4$, then factor-free languages a and aa, respectively, meet the bounds.

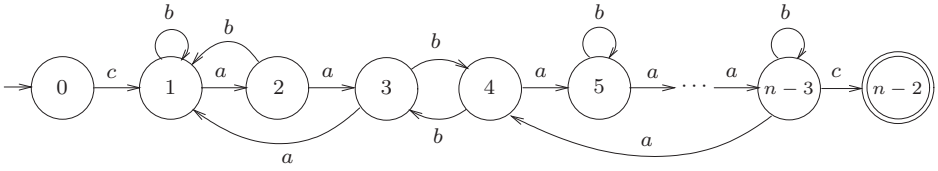

Figure 9: The ternary factor-free language meeting the bound $2^{n-3}+2$ for reversal.

If $n \geq 5$, then consider the language $L = cKc$, where K is a regular language over the alphabet $\{a, b\}$ with $\kappa(K) - 3$ meeting the upper bound 2^{n-3} for reversal [26, 39]. The quotient automaton of L without the empty state is shown in Figure 9.

By Proposition 1, language L is factor-free, and $\kappa(L)$. Since $\kappa(K^R) = 2^{n-3}$, there exists a set S of 2^{n-3} words over $\{a, b\}$ that define distinct quotients of language K^R. Then the quotients of cK^Rc by $2^{n-3}+2$ words ε, cw with $w \in S$, and cuc for some word u in K^R, are distinct as well. This gives $\kappa(L^R) = 2^{n-3}+2$. □

Theorem 7 (Reversal: Subword-Free Languages). *If L is a subword-free language over an alphabet Σ with $\kappa(L)$, where $n \geq 3$, then $\kappa(L^R) \leq 2^{n-3}+2$. The bound is tight if $|\Sigma| \geq 2^{n-3}-1$, but cannot be met for smaller alphabets. The bound cannot be met if L contains a word of length at least 3.*

Proof. Suppose L is a subword-free language. Let $\mathcal{D} = (Q, \Sigma, \delta, s, \{f\})$ be the quotient dfa of L with $Q = \{s, q_1, \ldots, q_{n-3}, f, e\}$ as the state set, where e and f correspond to the quotients $\varnothing$ and ε. Construct an nfa $\mathcal{D}^R$ for L^R, and consider the corresponding subset automaton.

The initial state of the subset automaton is $\{f\}$, and no other state contains f. Next, all the states containing s can be merged. As in Theorem 6, we get at most $2^{n-3}+2$ reachable states. If $\kappa(L^R) = 2^{n-3}+2$, then the set $\{q_1, q_2, \ldots, q_{n-3}\}$ must be reachable. Therefore there must exist a non-empty word v such that, for all q_i, we have $\delta(q_i, v) = f$. Now suppose there exists a word w in L such that $|w| > 2$. Let $w = abx$ where $a, b \in \Sigma$ and $x \in \Sigma^+$. Also suppose $\delta(s, a) = q_i$ and $\delta(q_i, b) = q_j$. Then we have $av, abv \in L$, showing that L is not subword-free, which is a contradiction. Hence, if any word in L has length at least 3, then $\kappa(L^R) < 2^{n-3}+2$. Now note that, if all the words in L have length at most 2, the only possible quotients of L^R are L^R, $(L^R)_a$ for all $a \in \Sigma$, ε, and $\varnothing$. Therefore $\kappa(L^R) \leq |\Sigma| + 3$, and the second claim follows.

Now consider tightness. If $n = 3$, then the bound is met by the unary subword-free language a. Let $n \geq 4$ and $\ell = 2^{n-3} - 1$. Also let $\Sigma = \{a_1, a_2, \ldots, a_\ell\}$, and let $S_1, S_2, \ldots, S_\ell$ be all the non-empty subsets of $\{1, 2, \ldots, n-3\}$. Now let

$$L^R = a_1\Big(\bigcup_{j \in S_1} a_j\Big) \cup a_2\Big(\bigcup_{j \in S_2} a_j\Big) \cup \cdots \cup a_\ell\Big(\bigcup_{j \in S_\ell} a_j\Big).$$

Since L^R only contains two-letter words, languages L^R and L are subword-free. The quotients of L^R are L^R, ε, $\varnothing$, and $(L^R)_{a_i} = \bigcup_{j \in S_i} a_j$ for $i = 1, 2, \ldots, \ell$.

Therefore $\kappa(L^R) = l + 3 = 2^{n-3} + 2$. But for L, the only possible and distinct quotients are L, ε, $\varnothing$, and L_{a_i} for $i = 1, 2, \ldots, n-3$. Thus $\kappa(L)$. □

7 Conclusions

Our results are summarized in Tables 1 and 2, where "B-, F-free" stands for bifix-free and factor-free, and "S-free" for subword-free. The bounds for operations on prefix-free languages are from [17, 21], on suffix-free languages from [13, 18, 25], and on regular languages from [31, 33, 32, 42]. For languages over a unary alphabet $\Sigma = \{a\}$, the concepts prefix-, suffix-, factor-, and subword-free coincide, and L is xfix-free with $\kappa(L)$ if and only if $L = \{a^{n-2}\}$.

In the case of subword-free languages the size of the alphabet cannot be decreased. In the other cases, whenever the size of the alphabet is greater than 2, we do not know whether or not the bounds are tight for smaller alphabets.

	$K \cup L, K \oplus L$	$\lvert\Sigma\rvert$	$K \cap L$	$\lvert\Sigma\rvert$	$K \setminus L$	$\lvert\Sigma\rvert$
free unary	$\max(m, n)$		m if m, 1 otherwise		m if $m \neq n$, 1 otherwise	
prefix	$mn - 2$	2	$mn - 2(m + n - 3)$	2	$mn - (m + 2n - 4)$	2
suffix	$mn - (m + n - 2)$	2	$mn - 2(m + n - 3)$	2	$mn - (m + 2n - 4)$	2
B-, F-free	$mn - (m + n)$	3	$mn - 3(m + n - 4)$	2	$mn - (2m + 3n - 9)$	2
S-free	$mn - (m + n)$	s_1	$mn - 3(m + n - 4)$	s_2	$mn - (2m + 3n - 9)$	s_3
regular	mn	2	mn	2	mn	2

Table 1: Complexities of Boolean operations on xfix-free languages; $s_1 = m + n - 3$, $s_2 = m + n - 7$, $s_3 = m + n - 6$.

	KL	$\lvert\Sigma\rvert$	L^*	$\lvert\Sigma\rvert$	L^R	$\lvert\Sigma\rvert$
free unary	$m + n - 2$		$n - 2$		n	
prefix-free	$m + n - 2$	1	n	2	$2^{n-2} + 1$	3
suffix-free	$(m - 1)2^{n-1} + 1$	3	$2^{n-2} + 1$	2	$2^{n-2} + 1$	3
B-, F-free	$m + n - 2$	1	$n - 1$	2	$2^{n-3} + 2$	3
S-free	$m + n - 2$	1	$n - 1$	2	$2^{n-3} + 2$	$2^{n-3} - 1$
regular	$(2m - 1)2^{n-1}$	2	$2^{n-1} + 2^{n-2}$	2	2^n	2

Table 2: Complexities of product, star, and reversal on xfix-free languages.

References

[1] Ang, T. and Brzozowski, J. Languages convex with respect to binary relations, and their closure properties. *Acta Cybernet.*, 19(2):445–464, 2009.

[2] Bassino, F., Giambruno, L., and Nicaud, C. Complexity of operations on cofinite languages. In López-Ortiz, A., editor, *Proceedings of the 9th Latin American Theoretical Informatics Symposium, (LATIN)*, volume 6034 of *LNCS*, pages 222–233. Springer, 2010.

[3] Berstel, J., Perrin, D., and Reutenauer, C. *Codes and Automata (Encyclopedia of Mathematics and its Applications)*. Cambridge University Press, 2010.

[4] Brzozowski, J. Derivatives of regular expressions. *J. ACM*, 11(4):481–494, 1964.

[5] Brzozowski, J. Quotient complexity of regular languages. *J. Autom. Lang. Comb.*, 15(1/2):71–89, 2010.

[6] Brzozowski, J. In search of the most complex regular languages. *Int. J. Found. Comput. Sc.*, 24(6):691–708, 2013.

[7] Brzozowski, J., Jirásková, G., and Li, B. Quotient complexity of ideal languages. *Theoret. Comput. Sci.*, 470:36–52, 2013.

[8] Brzozowski, J., Jirásková, G., Li, B., and Smith, J. Quotient complexity of bifix-, factor-, and subword-free regular languages. In Dömösi, P. and Szabolcs, I., editors, *Automata and Formal Languages, (AFL 2011)*, pages 123–137. Institute of Mathematics and Informatics, College of Nyíregyháza, Hungary, 2011.

[9] Brzozowski, J., Jirásková, G., and Zou, C. Quotient complexity of closed languages. *Theory Comput. Syst.*, 54:277–292, 2014.

[10] Brzozowski, J. and Liu, B. Quotient complexity of star-free languages. *Internat. J. Found. Comput. Sci.*, 26(6):1261–1276, 2012.

[11] Brzozowski, J. and Szykuła, M. Large aperiodic semigroups. `http://arxiv.org/abs/1401.0157`, Dec 2013.

[12] Câmpeanu, C., Culik II, K., Salomaa, K., and Yu, S. State complexity of basic operations on finite languages. In Boldt, O. and Jürgensen, H., editors, *Revised Papers from the 4th International Workshop on Automata Implementation, (WIA)*, volume 2214 of *LNCS*, pages 60–70. Springer, 2001.

[13] Cmorik, Roland and Jirásková, Galina. Basic operations on binary suffix-free languages. In Kotásek et al. [29], pages 94–102.

[14] Eom, H-S., Han, Yo-S., and Jirásková, G. State complexity of basic operations on non-returning regular languages. In Jürgensen and Reis [28], pages 54–65.

[15] Haines, L. H. On free monoids partially ordered by embedding. *J. Combin. Theory*, 6(1):94–98, 1969.

[16] Han, Yo-S. and Salomaa, K. State complexity of union and intersection of finite languages. *Internat. J. Found. Comput. Sci.*, 19(3):581–595, 2008.

[17] Han, Yo-S. and Salomaa, K. State complexity of basic operations on suffix-free regular languages. *Theoret. Comput. Sci.*, 410(27-29):2537–2548, 2009.

[18] Han, Yo-S., Salomaa, K., and Wood, D. Operational state complexity of prefix-free regular languages. In Ésik, Z. and Fülöp, Z., editors, *Automata, Formal Languages, and Related Topics*, pages 99–115. University of Szeged, Hungary, 2009.

[19] Iván, S. On state complexities of unions of binary factor-free languages. `http://arxiv.org/abs/1405.1107`, 2014.

[20] Jirásek, J. and Jirásková, G. Cyclic shift on prefix-free languages. In Bulatov, A. A. and Shur, A. M., editors, *CSR*, volume 7913 of *Lecture Notes in Computer Science*, pages 246–257. Springer, 2013.

[21] Jirásková, G. and Krausová, M. Complexity in prefix-free regular languages. In McQuillan, I., Pighizzini, G., and Trost, B., editors, *Proceedings of the 12th International Workshop on Descriptional Complexity of Formal Systems* (*DCFS*), pages 236–244. University of Saskatchewan, 2010.

[22] Jirásková, G. and Masopust, T. Complexity in union-free regular languages. *Int. J. Found. Comput. Sci.*, 22(7):1639–1653, 2011.

[23] Jirásková, G. and Masopust, T. On the state complexity of the reverse of $\mathcal{R}$- and $\mathcal{J}$-Trivial regular languages. In Jürgensen and Reis [28], pages 136–147.

[24] Jirásková, G. and Nagy, B. On union-free and deterministic union-free languages. In Baeten, J. C. M., Ball, T., and de Boer, F. S., editors, *IFIP TCS*, volume 7604 of *Lecture Notes in Computer Science*, pages 179–192. Springer, 2012.

[25] Jirásková, G. and Olejár, P. State complexity of union and intersection of binary suffix-free languages. In Bordihn, H., Freund, R., Holzer, M., Kutrib, M., and Otto, F., editors, *Proc. of the Workshop on Non-Classical Models for Automata and Applications* (*NCMA*), pages 151–166. Austrian Computer Society, 2009.

[26] Jirásková, G. and Šebej, J. Reversal of binary regular languages. *Theor. Comput. Sci.*, 449:85–92, 2012.

[27] Jürgensen, H. and Konstantinidis, S. Codes. In Rozenberg, G. and Salomaa, A., editors, *Handbook of Formal Languages, Volume 1: Word, Language, Grammar*, pages 511–607. Springer, 1997.

[28] Jürgensen, H. and Reis, R., editors. *Descriptional Complexity of Formal Systems - 15th International Workshop, DCFS 2013, London, ON, Canada, July 22-25, 2013. Proceedings*, volume 8031 of *Lecture Notes in Computer Science*. Springer, 2013.

[29] Kotásek, Zdenek, Bouda, Jan, Cerná, Ivana, Sekanina, Lukás, Vojnar, Tomás, and Antos, David, editors. *Mathematical and Engineering Methods in Computer Science - 7th International Doctoral Workshop, MEMICS 2011, Lednice, Czech Republic, October 14-16, 2011, Revised Selected Papers*, volume 7119 of *Lecture Notes in Computer Science*. Springer, 2012.

[30] Krausová, M. Prefix-free regular languages: Closure properties, difference, and left quotient. In Kotásek et al. [29], pages 114–122.

[31] Leiss, E. Succinct representation of regular languages by boolean automata. *Theoret. Comput. Sci.*, 13:323–330, 2009.

[32] Maslov, A. N. Estimates of the number of states of finite automata. *Dokl. Akad. Nauk SSSR*, 194:1266–1268 (Russian)., 1970. English translation: Soviet Math. Dokl. **11** (1970) 1373–1375.

[33] Mirkin, B. G. On dual automata. *Kibernetika (Kiev)*, 2:7–10 (Russian)., 1966. English translation: Cybernetics **2** (1966) 6–9.

[34] Perrin, D. Finite automata. In van Leewen, J., editor, *Handbook of Theoretical Computer Science*, volume B, pages 1–57. Elsevier, 1990.

[35] Pighizzini, G. and Shallit, J. Unary language operations, state complexity and Jacobsthal's function. *Internat. J. Found. Comput. Sci.*, 13:145–159, 2002.

[36] Shyr, H. J. *Free Monoids and Languages*. Hon Min Book Co, Taiwan, 2001.

[37] Shyr, H. J. and Thierrin, G. Hypercodes. *Inform. and Control*, 24:45–54, 1974.

[38] Thierrin, G. Convex languages. In Nivat, M., editor, *Automata, Languages and Programming*, pages 481–492. North-Holland, 1973.

[39] Šebej, J. Reversal of regular languages and state complexity. In Pardubská, D., editor, *Proc. 10th ITAT*, pages 47–54. Šafárik University, Košice, 2010.

[40] Yu, S. Regular languages. In Rozenberg, G. and Salomaa, A., editors, *Handbook of Formal Languages*, volume 1, pages 41–110. Springer, 1997.

[41] Yu, S. State complexity of regular languages. *J. Autom. Lang. Comb.*, 6:221–234, 2001.

[42] Yu, S., Zhuang, Q., and Salomaa, K. The state complexities of some basic operations on regular languages. *Theoret. Comput. Sci.*, 125:315–328, 1994.

Received 27th December 2013