Operations on Boolean and Alternating Finite Automata

Michal Hospodár, Galina Jirásková, and Ivana Krajňáková(✉)

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

Abstract. We investigate the descriptional complexity of basic regular operations on languages represented by Boolean and alternating finite automata. In particular, we consider the operations of difference, symmetric difference, star, reversal, left quotient, and right quotient, and get tight upper bounds $m + n, m + n, 2^n, 2^n, m$, and 2^m, respectively, for Boolean automata, and $m + n + 1, m + n, 2^n, 2^n, m + 1$, and $2^m + 1$, respectively, for alternating finite automata. To describe witnesses for symmetric difference, we use a ternary alphabet. All the remaining witnesses are defined over binary or unary alphabets that are shown to be optimal.

1 Introduction

The Boolean finite automata (BFAs) are generalization of nondeterministic finite automata (NFAs). In an NFA, the transition function maps any pair of state and input symbol to a subset of states. This subset can be viewed as disjunction of its states. We obtain a BFA by considering other Boolean functions on states as a result of the transition function. Alternating finite automata (AFAs) start from the only one initial state, wheares Boolean automata may start their computation in any Boolean function designated as the initial function.

Boolean automata recognize the class of regular languages [2,4]. Every n-state Boolean automaton can be simulated by 2^{2^n}-state deterministic finite automaton (DFA), or by $(2^n + 1)$-state NFA, and both upper bounds are tight already in the binary case [2,10].

Some of the constructions and upper bounds for elementary operations on alternating automata were introduced in [5]. The upper bound $2^m + n + 1$ for concatenation from [5] has been shown to be tight in [8]. Detailed results for the square on alternating and Boolean automata can be found in [12]. Tight upper bounds for union and intersection were shown in [10]. For star and reversal, the upper and lower bound provided in [10] differed by one.

Research supported by grant VEGA 2/0084/15 and grant APVV-15-0091. This work was conducted as a part of PhD study of Michal Hospodár and Ivana Krajňáková at the Faculty of Mathematics, Physics and Informatics of the Comenius University.

F. V. Fomin and V. V. Podolskii (Eds.): CSR 2018, LNCS 10846, pp. 181–193, 2018.
https://doi.org/10.1007/978-3-319-90530-3_16

In this paper we continue the study of the operational complexity on Boolean and alternating finite automata. We improve the results on star and reversal from [10] and provide exact complexity of these two operations. We also examine other regular operations: complementation, difference, symmetric difference, left and right quotient on both Boolean and alternating automata. We get the exact complexity for each operation on both BFAs and AFAs. All our witness languages are defined over a small fixed alphabet which is optimal in most of the cases.

2 Preliminaries

Let Σ be a finite alphabet of symbols. Then Σ^* denotes the set of words over Σ including the empty word ε. A language is any subset of Σ^*. The cardinality of a finite set A is denoted by $|A|$, and its power-set by 2^A. The reader may refer to [7,17,18] for details.

A *nondeterministic finite automaton* (NFA) is a quintuple $A = (Q, \Sigma, \circ, I, F)$, where Q is a finite set of states, Σ is a finite non-empty alphabet, $\circ : Q \times \Sigma \to 2^Q$ is the transition function which is naturally extended to the domain $2^Q \times \Sigma^*$, $I \subseteq Q$ is the set of initial states, and $F \subseteq Q$ is the set of final states. The *language accepted by* A is the set $L(A) = \{w \in \Sigma^* \mid I \circ w \cap F \neq \emptyset\}$. For a symbol a, we say that (p, a, q) is a transition in NFA A if $q \in p \circ a$, and the state q has an in-transition on a. For a word w, we write $p \xrightarrow{w} q$ if $q \in p \circ w$.

An NFA A is *deterministic* (DFA) if $|I| = 1$ and $|q \circ a| = 1$ for each q in Q and each a in Σ; so all DFAs in this paper are assumed to be complete. We write $p \cdot a = q$ instead of $p \circ a = \{q\}$ in such a case. The *state complexity* of a regular language L, $\operatorname{sc}(L)$, is the smallest number of states in any DFA for L. A state q of a DFA is called *sink state* if $q \cdot a = q$ for each a in Σ.

For unary DFAs we use the Nicaud's notation [15]. For two integers ℓ and n such that $0 \leq \ell \leq n-1$ and a subset F of $\{0, \ldots, n-1\}$, $A = (n, \ell, F)$ is the unary automaton whose set of states is $Q = \{0, \ldots, n-1\}$ and the transition function is given by $q \cdot a = q + 1$ if $0 \leq q \leq n-2$ and $(n-1) \cdot a = \ell$. The initial state of this automaton is 0 and its set of final states is F.

Every NFA $A = (Q, \Sigma, \circ, I, F)$ can be converted to an equivalent DFA $\mathcal{D}(A) = (2^Q, \Sigma, \cdot, I, F')$, where $S \cdot a = S \circ a$ for each S in 2^Q and a in Σ and $F' = \{R \in 2^Q \mid R \cap F \neq \emptyset\}$. We call the DFA $\mathcal{D}(A)$ the *subset automaton* of the NFA A. The subset automaton may not be minimal since some of its states may be unreachable or equivalent to other states.

To prove distinguishability of the states of the subset automaton, the following notions and observations are useful. A state q of an NFA A is called *uniquely distinguishable* if there is a word w which is accepted by A from and only from the state q, that is $p \circ w \cap F \neq \emptyset$ if and only if $p = q$. A transition (p, a, q) is called a *unique in-transition* if there is no state r such that $r \neq p$ and (r, a, q) is a transition in A. A state q is *uniquely reachable* from a state p if there exists a sequence of unique in-transitions (q_i, a, q_{i+1}) for $i = 0, 1, \ldots, k$ such that $q_0 = p$ and $q_{k+1} = q$.

Proposition 1 [1, Propositions 14 and 15]**.** *Let A be an* NFA *and $\mathcal{D}(A)$ be the corresponding subset automaton.*

(a) If two subsets of $\mathcal{D}(A)$ differ in a uniquely distinguishable state of A, then they are distinguishable.
(b) If a state q of A is uniquely distinguishable and uniquely reachable from a state p, then the state p is uniquely distinguishable as well.
(c) If there is a uniquely distinguishable state of A which is uniquely reachable from any other state of A, then every state of A is uniquely distinguishable.
(d) If every state of A is uniquely distinguishable, then the subset automaton $\mathcal{D}(A)$ does not have equivalent states.

□

Let K and L be languages over an alphabet Σ. The *difference* and *symmetric difference* of K and L are the languages $K \backslash L = \{w \in K \mid w \notin L\}$ and $K \oplus L = \{w \in K \mid w \notin L\} \cup \{w \in L \mid w \notin K\}$, respectively. If languages K and L are accepted by DFAs $A = (Q_A, \Sigma, \cdot_A, s_A, F_A)$ and $B = (Q_B, \Sigma, \cdot_B, s_B, F_B)$, then the language $K \cap L$ is accepted by the *product automaton* $A \times B = (Q_A \times Q_B, \Sigma, \cdot, (s_A, s_B), F_A \times F_B)$ where $(p, q) \cdot a = (p \cdot_A a, q \cdot_B a)$. For the remaining Boolean operations we only need to change the set of final states in the product automaton. For union, difference, symmetric difference the set of final states is $(F_A \times Q_B) \cup (Q_A \times F_B)$, $F_A \times (Q_B \backslash F_B)$, $(F_A \times (Q_B \backslash F_B)) \cup ((Q_A \backslash F_A) \times F_B)$, respectively.

The reverse of a word is defined as $\varepsilon^R = \varepsilon$ and $(wa)^R = aw^R$ for each symbol a and word w. The reverse of a language L is the language $L^R = \{w^R \mid w \in L\}$. The reverse of an NFA A is an NFA A^R obtained from A by reversing all the transitions and by swapping the roles of initial and final states. The NFA A^R recognizes the reverse of $L(A)$.

The *concatenation of* K and L is the language $KL = \{uv \mid u \in K \text{ and } v \in L\}$. The *square* of a language L is the language $L^2 = LL$. The *right quotient* of K by L is the language $KL^{-1} = \{x \in \Sigma^* \mid xy \in K \text{ for some } y \in L\}$. The *left quotient* of K by L is the language $L^{-1}K = \{x \in \Sigma^* \mid yx \in K \text{ for some } y \in L\}$.

A *Boolean finite automaton* (BFA) is a quintuple $A = (Q, \Sigma, \delta, g_s, F)$, where Q is a finite non-empty set of states, $Q = \{q_1, \ldots, q_n\}$, Σ is an input alphabet, δ is the transition function that maps $Q \times \Sigma$ into the set $\mathcal{B}_n$ of Boolean functions with variables $\{q_1, \ldots, q_n\}$, $g_s \in \mathcal{B}_n$ is the initial Boolean function, and $F \subseteq Q$ is the set of final states. The transition function δ can be extended to the domain $\mathcal{B}_n \times \Sigma^*$ as follows: For all g in $\mathcal{B}_n$, a in Σ, and w in Σ^*, we have $\delta(g, \varepsilon) = g$; if $g = g(q_1, \ldots, q_n)$, then $\delta(g, a) = g(\delta(q_1, a), \ldots, \delta(q_n, a))$; $\delta(g, wa) = \delta(\delta(g, w), a)$. Next, let $f = (f_1, \ldots, f_n)$ be the Boolean vector with $f_i = 1$ iff $q_i \in F$. The language accepted by the BFA A is the set $L(A) = \{w \in \Sigma^* \mid \delta(g_s, w)(f) = 1\}$.

A Boolean finite automaton is called *alternating* (AFA) if the initial function is a projection $g(q_1, \ldots, q_n) = q_i$. For details, we refer to [2,5,10,13,17,18].

The *Boolean (alternating) state complexity* of L, $\operatorname{bsc}(L)(\operatorname{asc}(L))$, is the smallest number of states in any BFA (AFA) for L. It is known that a language L is accepted by an n-state BFA (AFA) if and only if the language L^R is accepted by an 2^n-state DFA (with 2^{n-1} final states). Since this is the crucial observation used later in the paper, we state it in the next two lemmas and provide proof ideas here.

Lemma 2 (cf. [5, Theorem 4.1, **Corollary** 4.2] **and** [10, **Lemma** 1]). *Let L be a language accepted by an n-state* BFA *(*AFA*). Then the reversal L^R is accepted by a* DFA *of 2^n states (of which 2^{n-1} are final).*

Proof (Proof Idea). Let $A = (\{q_1, q_2, \dots, q_n\}, \Sigma, \delta, g_s, F)$ be an n-state BFA for L. Construct a 2^n-state NFA $A' = (\{0,1\}^n, \Sigma, \delta', S, \{f\})$, where

– for every $u = (u_1 \dots, u_n) \in \{0,1\}^n$ and every $a \in \Sigma$, $\delta'(u, a) = \{u' \in \{0,1\}^n \mid \delta(q_{i,a})(u') = u_i \text{ for } i = 1, \dots, n\}$;
– $S = \{(b_1, \dots, b_n) \in \{0,1\}^n \mid g_s(b_1, \dots, b_n) = 1\}$;
– $f = (f_1, \dots, f_n) \in \{0,1\}^n$ with $f_i = 1$ iff $q_i \in F$.

Then $L(A) = L(A')$ and $(A')^R$ is deterministic. Moreover if A is an AFA then A' has 2^{n-1} initial states. It follows that L^R is accepted by a DFA with 2^n states, of which 2^{n-1} are final if A is an AFA. □

Lemma 3 (cf. [10, **Lemma** 2]). *Let L^R be accepted by a* DFA *A of 2^n states (of which 2^{n-1} are final). Then L is accepted by an n-state* BFA *(*AFA*).*

Proof (Proof Idea). Consider 2^n-state NFA A^R for L which has exactly one final state and the set of initial states S (and $|S| = 2^{n-1}$). Let the state set Q of A^R be $\{0, 1, \dots, 2^n - 1\}$ with final state k and the initial set S ($S = \{2^{n-1}, \dots, 2^n - 1\}$). Let δ be the transition function of A^R. Moreover, for every $a \in \Sigma$ and for every $i \in Q$, there is exactly one state j such that j goes to i on a in A^R. For a state $i \in Q$, let $\text{bin}(i) = (b_1, \dots, b_n)$ be the binary n-tuple such that $b_1 b_2 \cdots b_n$ is the binary notation of i on n digits with leading zeros if necessary.

Let us define an n-state BFA $A' = (Q', \Sigma, \delta', g_s, F')$, where $Q' = \{q_1, \dots, q_n\}$, $F' = \{q_\ell \mid \text{bin}(k)_\ell = 1\}$, and $g_s(\text{bin}(i)) = 1$ iff $i \in S$ ($g_s = q_1$). We define δ' to suffice the condition: for each i in Q and a in Σ, $(\delta'(q_1, a), \dots, \delta'(q_n, a))(\text{bin}(i)) = \text{bin}(j)$ where $i \in \delta(j, a)$. Then $L(A') = L(A^R)$. □

As a corollary of the previous two lemmas, we get the following results.

Corollary 4. *If L is a regular language, then* $\text{bsc}(L) \geq \lceil \log(\text{sc}(L^R)) \rceil$ *and* $\text{asc}(L) \geq \lceil \log(\text{sc}(L^R)) \rceil$. □

Corollary 5. *Let L be a unary language. Then L is accepted by an n-state* BFA *(*AFA*) if and only if L is accepted by a 2^n-state* DFA *(with 2^{n-1} final states).* □

Now we prove several propositions which we use later in our paper.

Proposition 6. *If L is accepted by an n-state* BFA*, then L is accepted by an $(n+1)$-state* AFA*.*

Proof. Let a language L be accepted by an n-state BFA $(Q, \Sigma, \delta, g, F)$. Let $A = (Q \cup \{s\}, \Sigma, \delta', s, F')$ where $s \notin Q$, $\delta'(q, a) = \delta(q, a)$ if $q \in Q$ and $\delta'(q, a) = \delta(g, a)$ if $q = s$; $F' = F$ if $\varepsilon \notin L$ and $F' = F \cup \{s\}$ if $\varepsilon \in L$. Then A is an $(n + 1)$-state AFA for L. □

Proposition 7. *Let K and L be languages over Σ. Then*

(a) $(KL^{-1})^R = (L^R)^{-1}K^R$;
(b) $(L^{-1}K)^R = K^R(L^R)^{-1}$.

□

Proposition 8. *Let a non-empty language L be accepted by an n-state* DFA. *Then L^* is accepted by a 2^n-state* DFA *with half of the states final.*

Proof. Let L be accepted by an n-state DFA $A = (Q, \Sigma, \cdot, s, F)$. If the initial state is the only final state in A, then $L^* = L$, and we may add final and non-final unreachable sink states to get the desired automaton. Otherwise there is a final state q_F such that $q_F \neq s$. Construct an NFA N for L^* from A as follows:

(a) add the transition (q, a, s) whenever $q \cdot a \in F$;
(b) add a new initial and final state q_0;
(c) the initial states of N are s and q_0 and the set of final states is $F \cup \{q_0\}$.

In the corresponding subset automaton $\mathcal{D}(N)$ the initial subset is $\{q_0, s\}$ and any other reachable subset S is a non-empty subset of Q such that $S \cap F \neq \emptyset$ implies $s \in S$. By the construction above every set S such that $q_F \in S$ and $s \notin S$ is unreachable. That means that there are at most $1 + 2^n - 1 - 2^{n-2} = \frac{3}{4}2^n$ reachable sets in $\mathcal{D}(N)$. Let us show that in the minimal DFA for L^* the number of non-final states as well as the number of final states is at most 2^{n-1}. The non-final subsets in $\mathcal{D}(N)$ must not contain the state q_F, so there are at most 2^{n-1} of them. Next the initial subset $\{q_0, s\}$ is final and any other final subset must contain the state s. This gives at most $1 + 2^{n-1}$ subsets. However, if $s \in F$ then $\{q_0, s\}$ and $\{s\}$ are equivalent, and if $s \notin F$ then $\{s\}$ is non-final. Therefore the minimal DFA for L^* has at most 2^{n-1} final states. To obtain 2^n-state DFA we may add some unreachable sink states. Since the number of final and non-final states are at most 2^{n-1} it is possible to achieve that exactly half of the states would be final and the other half non-final in the resulting 2^n-state DFA. □

Proposition 9. *Let $m, n \geq 2$ and* $\gcd(m, n) = 1$. *Let K and L be unary regular languages accepted by deterministic finite automata $A = (m, 0, \{0\})$ and $B = (n, 0, \{1, 2, \ldots, n-1\})$, respectively. Then* $\mathrm{sc}(K \oplus L) = mn$.

Proof. Since symmetric difference is a commutative operation, we may assume that $m < n$. Denote $Q_A = \{0, 1, \ldots, m - 1\}$, $Q_B = \{0, 1, \ldots, n - 1\}$. Consider the product automaton $A \times B = (Q_A \times Q_B, \{a\}, \cdot, (0, 0), F)$ where the set of final states is $F = \{(0, 0)\} \cup \{1, 2, \ldots, m - 1\} \times \{1, 2, \ldots, n - 1\}$. Since $\gcd(m, n) = 1$, every state of the product automaton is reachable. To prove distinguishability, let p and q be two distinct states of the product automaton. Then there is an integer $k \geq 0$ such that $p \cdot a^k = (m - 1, 0)$ and $q \cdot a^k = q'$ where $q' \neq (m - 1, 0)$. We have three cases:

(a) $q' \in F$. Then a^k distinguishes p and q since $(m-1,0) \notin F$.
(b) $q' = (0, n-1)$. Then $a^k a^m$ distinguishes p and q since

$$p \xrightarrow{a^k} (m-1,0) \xrightarrow{a^m} (m-1,m) \in F,$$
$$q \xrightarrow{a^k} (0,n-1) \xrightarrow{a} (1,0) \xrightarrow{a^{m-1}} (0,m-1) \notin F; \text{ recall that } m < n.$$

(c) q' is a non-final state different from $(0, n-1)$. Then $a^k a$ distinguishes p and q since $(m-1,0) \cdot a \notin F$ and $q' \cdot a \in F$.

Hence all the states of the product automaton are reachable and pairwise distinguishable. This means that $\mathrm{sc}(K \oplus L) = mn$. □

3 Operations on Boolean and Alternating Automata

In this section we investigate the descriptional complexity of basic regular operations on languages represented by Boolean and alternating automata. We start with the complementation operation and we show that a language and its complement have the same complexity.

Theorem 10 (Complementation). *Let L be a regular language. Then we have* $\mathrm{asc}(L) = \mathrm{asc}(L^c)$ *and* $\mathrm{bsc}(L) = \mathrm{bsc}(L^c)$.

Proof. Let L be accepted by a minimal n-state BFA (AFA). Then the language L^R is accepted by a 2^n-state DFA (with half of the states final) by Lemma 2. This means that $(L^R)^c$ is accepted by a 2^n state DFA (with half of the states final) since we only interchange final and non-final states in the DFA for L^R. Next $(L^R)^c = (L^c)^R$. Therefore L^c is accepted by an n-state BFA (AFA) by Lemma 3. Hence $\mathrm{asc}(L^c) \le n$ and $\mathrm{bsc}(L^c) \le n$. Moreover we cannot have $\mathrm{asc}(L^c) < n$ because after another complementation we would get $\mathrm{asc}(L) < n$. The argument for $\mathrm{bsc}(L^c)$ is the same. □

We continue with the star operation. We improve the results from [10, Theorems 8, 9] where upper and lower bounds differed by one. We get tight upper bound 2^n for both BFAs and AFAs as a corollary of the next theorem.

Theorem 11 (Star). *Let $n \ge 2$.*

(a) If L is accepted by an n-state BFA*, then L^* is accepted by a 2^n-state* AFA*.*
(b) There exists a language L accepted by an n-state AFA *such that every* BFA *for L^* has at least 2^n states.*

Proof

(a) Let L be accepted by an n-state BFA. Then L^R is accepted by a 2^n-state DFA by Lemma 2. By Propostion 8, $(L^R)^*$ is accepted by a 2^{2^n}-state DFA with half of the states final. Next $(L^R)^* = (L^*)^R$. This means that L^* is accepted by a 2^n-state AFA by Lemma 3.

(b) Let L^R be the Palmovský's witness language for star [16] with 2^n states and 2^{n-1} final states shown in Fig. 1. By Lemma 3 the language L is accepted by an n-state AFA. By [16, Proof of Theorem 4.4] $\mathrm{sc}((L^R)^*) = 2^{2^n-1} + 2^{2^n-1-2^{n-1}} = 2^{2^n-1}(1+2^{-2^{n-1}})$. Since $(L^R)^* = (L^*)^R$ we get $\mathrm{bsc}(L^*) \ge \lceil \log(\mathrm{sc}((L^*)^R)) \rceil = 2^n$ by Corollary 4. □

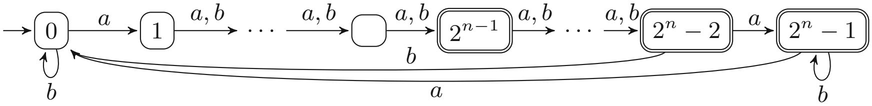

Fig. 1. The reverse of a binary witness for star on BFAs and AFAs.

In what follows we use Lemmas 2, 3 and Corollary 4 without citing them again and again. The next theorem provides tight upper bounds on the complexity of difference, symmetric difference, reversal, and right and left quotient on languages represented by Boolean finite automata.

Theorem 12 (Operations on BFAs). *Let K and L be (regular) languages over an alphabet Σ accepted by an m-state and n-state* BFA*, respectively. Then*

(a) $\mathrm{bsc}(K \setminus L) \leq m+n$*, and the bound is tight if* $|\Sigma| \geq 2$*;*
(b) $\mathrm{bsc}(K \oplus L) \leq m+n$*, and the bound is tight if* $|\Sigma| \geq 3$*;*
(c) $\mathrm{bsc}(L^R) \leq 2^n$*, and the bound is tight if* $|\Sigma| \geq 2$*;*
(d) $\mathrm{bsc}(KL^{-1}) \leq 2^m$*, and the bound is tight if* $|\Sigma| \geq 2$*;*
(e) $\mathrm{bsc}(L^{-1}K) \leq m$*, and the bound is tight if* $|\Sigma| \geq 1$*.*

Proof. Let $A = (Q_A, \Sigma, \delta_A, g_A, F_A)$ be an m-state BFA for the language K and $B = (Q_B, \Sigma, \delta_B, g_B, F_B)$ be an n-state BFA for L with $Q_A \cap Q_B = \emptyset$.

(a) The language $K \setminus L$ is accepted by BFA $(Q_A \cup Q_B, \Sigma, \delta, g_A \wedge \overline{g_B}, F_A \cup F_B)$, where $\delta = \delta_A$ on Q_A and $\delta = \delta_B$ on Q_B. Thus $\mathrm{bsc}(K \setminus L) \leq m+n$. For tightness, let K and L be binary witness languages for intersection on BFAs described in [10, Proof of Theorem 2]. Then K and L^c are witnesses for difference since $K \setminus L^c = K \cap L$.

(b) The symmetric difference $K \oplus L$ is accepted by BFA

$$(Q_A \cup Q_B, \Sigma, \delta, (g_A \wedge \overline{g_B}) \vee (\overline{g_A} \wedge g_B), F_A \cup F_B)$$

where $\delta = \delta_A$ on Q_A and $\delta = \delta_B$ on Q_B. Thus $\mathrm{bsc}(K \oplus L) \leq m+n$. For tightness, let K^R and L^R be the languages accepted by 2^m-state and 2^n-state DFAs with half of states final shown in Fig. 2. Then K and L are accepted by m-state and n-state BFAs. In the product automaton, each state (i, j) is reached by $a^i b^j$. Two (non-)final states are distinguished by c if they are in different quadrants and by a word in $a^* + b^*$ otherwise. So we get $\mathrm{sc}(K^R \oplus L^R) = 2^{m+n}$. Next $K^R \oplus L^R = (K \oplus L)^R$. Therefore $\mathrm{bsc}(K \oplus L) \geq m+n$.

(c) The language L^R is accepted by 2^n-state DFA, the special case of BFA. For tightness, let L^R be the Šebej's binary witness language for reversal [11] accepted by a DFA with 2^n states. Then L is accepted by an n-state BFA. By [11, Proof of Theorem 5] $\mathrm{sc}((L^R)^R) = 2^{2^n}$ and therefore $\mathrm{bsc}(L^R) \geq 2^n$.

(d) If K and L are accepted by an m-state and n-state BFA, respectively, then K^R and L^R are accepted by a 2^m-state and 2^n-state DFA, respectively. By Proposition 7 $(KL^{-1})^R = (L^R)^{-1}K^R$ and by [19, Theorem 4.1] $\mathrm{sc}((L^R)^{-1}K^R) \leq 2^{2^m} - 1$. It follows that $\mathrm{bsc}(KL^{-1}) \leq 2^m$. For tightness, let $L = \Sigma^*$ and K be the language accepted by the DFA shown in Fig. 3. Then $\mathrm{bsc}(K) \leq m$ and

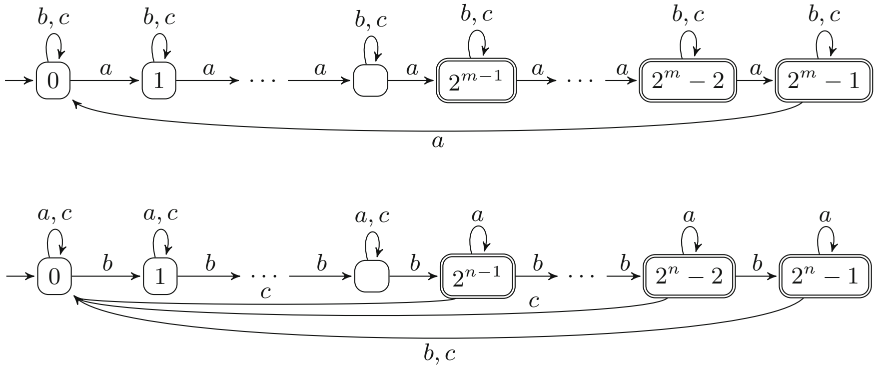

Fig. 2. The reverses of ternary witnesses for symmetric difference on BFAs.

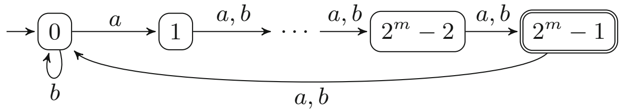

Fig. 3. The reverse of a binary witness for right quotient (by Σ^*) on BFAs.

$\mathrm{bsc}(L) \le n$. Next $(KL^{-1})^R = (\Sigma^*)^{-1}K^R$ and by [19, Proof of Theorem 4.1] $\mathrm{sc}((\Sigma^*)^{-1}K^R) = 2^{2^m} - 1$. Therefore $\mathrm{bsc}(KL^{-1}) \ge 2^m$.

(e) Since $(L^{-1}K)^R = K^R(L^R)^{-1}$ and $\mathrm{sc}(K^R(L^R)^{-1}) \le 2^m$ [19, p. 323], we get $\mathrm{bsc}(L^{-1}K) \le m$. For tightness, let $K = \{a^i \mid 2^{m-1}-1 \le i \le 2^m-2\}$ and $L = a^*$. Then $\mathrm{bsc}(K) \le m$ and $\mathrm{bsc}(L) \le n$. Next $K^R(a^*)^{-1} = \{a^i \mid 0 \le i \le 2^m - 2\}$, so $\mathrm{sc}(K^R(a^*)) = 2^m$. Therefore $\mathrm{bsc}(L^{-1}K) \ge m$.

In the next theorem we study the complexities of same operations on languages represented by alternating finite automata. Note that while the complexities of intersection, union, and difference on AFAs exceed those on BFAs by one, the complexity of symmetric difference on AFAs and BFAs is the same.

Theorem 13 (Operations on AFAs). *Let K and L be (regular) languages over an alphabet Σ accepted by an m-state and n-state* AFA*, respectively. Then*

(a) $\mathrm{asc}(K \setminus L) \le m + n + 1$*, and the bound is tight if* $|\Sigma| \ge 2$*;*
(b) $\mathrm{asc}(K \oplus L) \le m + n$*, and the bound is tight if* $|\Sigma| \ge 3$*;*
(c) $\mathrm{asc}(L^R) \le 2^n$*, and the bound is tight if* $|\Sigma| \ge 2$*;*
(d) $\mathrm{asc}(KL^{-1}) \le 2^m + 1$*, and the bound is tight if* $|\Sigma| \ge 2$*;*
(e) $\mathrm{asc}(L^{-1}K) \le m + 1$*, and the bound is tight if* $|\Sigma| \ge 1$*.*

Proof

(a) Since every AFA is BFA we get $\mathrm{bsc}(K \setminus L) \le m + n$ by Theorem 12(a). Therefore $\mathrm{asc}(K \setminus L) \le m+n+1$. For tightness, let K and L be the binary witness languages for intersection on AFAs described in [10, Proof of Theorem 3]. Then K and L^c are witnesses for difference since $\mathrm{asc}(K \setminus L^c) = \mathrm{asc}(K \cap L) = m + n + 1$.

(b) If K and L are accepted by m-state and n-state AFAs, then K^R and L^R are accepted by 2^m-state and 2^n-state DFAs with half of the states final. It follows that $K^R \oplus L^R$ is accepted by a product automaton of 2^{m+n} states and half of them are final. Therefore $K \oplus L$ is accepted by $(m+n)$-state AFA. For tightness, let K^R and L^R be the languages accepted by 2^m-state and 2^n-state DFAs with half of the states final shown in Fig. 2. Then K and L are accepted by m-state and n-state AFAs. As shown in Theorem 12(b) every BFA for $K \oplus L$ has at least $m+n$ states. Therefore $\operatorname{asc}(K \oplus L) \geq m+n$.

(c) If L is accepted by an n-state AFA, then L^R is accepted by 2^n-state DFA. Every DFA is a special case of AFA. Therefore AFA for language L^R has 2^n states. For tightness, let L^R be the language accepted by 2^n-state Šebej's automaton in which half of the states are final shown in Fig. 4. By [11, Proof of Theorem 5] we have $\operatorname{sc}((L^R)^R) = 2^{2^n}$; notice that any nontrivial number of final states does not matter since the subset automaton of NFA for $(L^R)^R$ does never have equivalent states [11, Proposition 3]. Hence $\operatorname{asc}(L^R) \geq 2^n$ by Corollary 4.

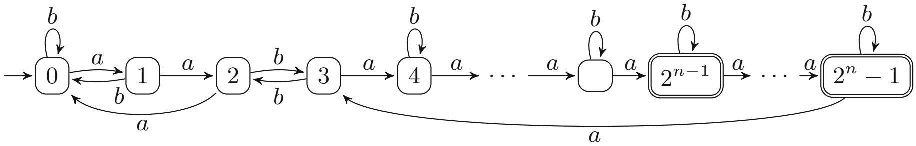

Fig. 4. The reverse of a binary witness for reversal on AFAs.

(d) By Propostion 6 and Theorem 12(d) we get $\operatorname{asc}(KL^{-1}) \leq \operatorname{bsc}(KL^{-1}) + 1 \leq 2^m + 1$. To prove tightness, let $L = \Sigma^*$ and K^R be the language accepted by the DFA A shown in Fig. 5 in which half of the states are final. Then $\operatorname{asc}(K) \leq m$ and $\operatorname{asc}(L) \leq n$. Next $(KL^{-1})^R = (\Sigma^*)^{-1}K^R$. Let us show that $\operatorname{sc}((\Sigma^*)^{-1}K^R) = 2^{2^m} - 1$. Construct an NFA N for $(\Sigma^*)^{-1}K^R$ from the DFA A by making all the states initial. Every non-empty subset in the corresponding subset automaton is reachable as it was shown in [19, Proof of Theorem 4.1]. To prove distinguishability, notice that the state 1 is uniquely distinguishable by the word b^{2^m-2}, and it is uniquely reachable in N from any other state through the unique in-transitions $2 \xrightarrow{a} 3 \xrightarrow{a} \cdots \xrightarrow{a} 2^m{-}1 \xrightarrow{a} 0 \xrightarrow{a} 1$. By Proposition 1, all states of the subset automaton are pairwise distinguishable. The number of final states in the subset automaton is $2^{2^m} - 2^{2^{m-1}}$, which is greater than 2^{2^m-1}. Therefore by Lemma 2 we get $\operatorname{asc}(KL^{-1}) \geq 2^m + 1$.

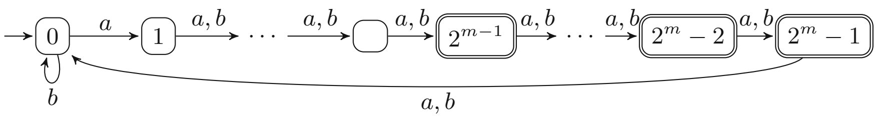

Fig. 5. The reverse of a binary witness for right quotient (by Σ^*) on AFAs.

(e) By Proposition 6 and Theorem 12(e) $\mathrm{asc}(L^{-1}K) \le \mathrm{bsc}(L^{-1}K) + 1 \le m + 1$. To get tightness, consider the same two languages as in Theorem 12(e). Notice that the minimal DFA for $K^R(a^*)^{-1}$ has more than 2^{m-1} final states. □

In the next theorem we study the complexity of basic regular operations on unary languages represented by Boolean finite automata.

Theorem 14 (Unary BFAs). *Let $n \ge 2$ and K and L be unary languages accepted by an m-state and n-state* BFA*, respectively. Then*

(a) $\mathrm{bsc}(K \cap L) \le m + n$*, and the bound is tight if* $\gcd(m, n) = 1$*;*
(b) $\mathrm{bsc}(K \cup L) \le m + n$*, and the bound is tight if* $\gcd(m, n) = 1$*;*
(c) $\mathrm{bsc}(K \setminus L) \le m + n$*, and the bound is tight if* $\gcd(m, n) = 1$*;*
(d) $\mathrm{bsc}(K \oplus L) \le m + n$*, and the bound is tight if* $\gcd(m, n) = 1$*;*
(e) $\mathrm{bsc}(L^R) = \mathrm{bsc}(L)$*;*
(f) $\mathrm{bsc}(L^*) \le 2n$ *and the bound is tight;*
(g) $\mathrm{bsc}(KL^{-1}) \le m$*, and the bound is tight.*

Proof. Let unary languages K and L be accepted by m-state and n-state BFA, respectively. Then K and L are accepted by 2^m-state and 2^n-state DFA, respectively, by Corollary 5, and the languages $K \cap L$, $K \cup L$, $K \setminus L$, $K \oplus L$ are accepted by a $2^m 2^n$-state product automaton. This gives upper bounds $m + n$ in cases (a)–(d). To prove tightness for intersection, let $K = (a^{2^m})^*$ and $L = (a^{2^n-1})^*$. Then K and L are accepted by a 2^m-state and 2^n-state DFA, respectively, so by an m-state and n-state BFA, respectively. Since $\gcd(2^m, 2^n - 1) = 1$, we have $\mathrm{sc}(K \cap L) = 2^m(2^n - 1)$. This means that $\mathrm{bsc}(K \cap L) \ge \lceil \log(2^m(2^n - 1)) \rceil = m + n$. For union, we may use the languages K^c and L^c, since $K^c \cup L^c = (K \cap L)^c$ and a language and its complement have the same Boolean state complexity. Similarly, for difference we use the languages K and L^c. For symmetric difference, let us consider unary languages K and L accepted by automata $A = (2^m, 0, \{0\})$ and $B = (2^n - 1, 0, \{1, 2, \ldots, 2^n - 2\})$. By Proposition 9 $\mathrm{sc}(K \oplus L) = 2^m(2^n - 1)$. It follows that $\mathrm{bsc}(K \oplus L) \ge \lceil \log(2^m(2^n - 1)) \rceil = m + n$.

(e) The equality follows from the fact that $L = L^R$ in the unary case.

(f) The state complexity of the star operation in the unary case is $(n-1)^2+1$ [3,19]. If a unary language L is accepted by an n-state BFA then L is accepted by a 2^n-state DFA. This means that L^* is accepted by a DFA of at most $(2^n-1)^2+1$ states, so by a DFA of at most 2^{2n} states. Therefore $\mathrm{bsc}(L^*) \le 2n$. For tightness, let L be the unary language accepted by the DFA $(2^n, 0, \{2^n - 1\})$ meeting the upper bound for star [19, Theorem 5.3]. Then L is accepted by an n-state BFA and $\mathrm{bsc}(L^*) \ge \lceil \log(\mathrm{sc}(L^*)) \rceil = \lceil \log((2^n - 1)^2 + 1) \rceil = 2n$.

(g) In the unary case, $KL^{-1} = L^{-1}K$. In Theorem 12(e) we proved that $\mathrm{bsc}(L^{-1}K) \le m$ and we provided a unary witness. □

Recall that by Proposition 6 $\mathrm{asc}(L) \le \mathrm{bsc}(L) + 1$. Therefore as a corollary of the previous theorem we get the following upper bounds.

Corollary 15 (Unary AFAs). *Let $n \ge 2$ and K and L be unary languages accepted by an m-state and n-state* AFA*, respectively. Then*

(a) $\text{asc}(K \cap L) \leq m + n + 1$;
(b) $\text{asc}(K \cup L) \leq m + n + 1$;
(c) $\text{asc}(K \setminus L) \leq m + n + 1$;
(d) $\text{asc}(L^R) = \text{asc}(L)$;
(e) $\text{asc}(L^*) \leq 2n + 1$;
(f) $\text{asc}(KL^{-1}) \leq m + 1$.

We are not able to prove the tightness since the complexity of operations on unary DFAs with half of the states final is not known. The previous theorem and its corollary imply that a binary alphabet for some of our witness languages is optimal in the sense that it cannot be reduced to a unary alphabet.

4 Conclusions

We investigated the descriptional complexity of basic regular operations on languages represented by Boolean and alternating finite automata. We considered the operations of complementation, star, difference, symmetric difference, reversal, and left and right quotient. For each operation we obtained the tight upper bound on its complexity on both Boolean and alternating automata.

Our results are summarized in Table 1. The table also shows the size of alphabet used for describing witness languages, and compares our results to the known results for deterministic [11,14,19] and nondeterministic finite automata from [6,9]. The results for intersection and union on Boolean and alternating automata are from [10]. Notice that the complexity of intersection, union, and difference on alternating automata is $m+n+1$ while the complexity of symmetric difference is $m+n$. Except for ternary witnesses for symmetric difference, all the other provided witnesses are defined over a binary or unary alphabets and, moreover, a binary alphabet for the witness languages for star, reversal, and right quotient on BFAs and AFAs is optimal in the sense that it cannot be reduced to a unary alphabet.

Table 1. The complexity of operations on languages represented by BFAs, AFAs, DFAs, NFAs. The results for DFAs are from [11,14,19], the results for NFAs are from [6,9], and the results for intersection and union on BFAs and AFAs are from [10].

	BFA	$\|\Sigma\|$	AFA	$\|\Sigma\|$	DFA	$\|\Sigma\|$	NFA	$\|\Sigma\|$
Complement	n	1	n	1	n	1	2^n	2
Intersection	$m+n$	2	$m+n+1$	2	mn	2	mn	2
Union	$m+n$	2	$m+n+1$	2	mn	2	$m+n+1$	2
Difference	$m+n$	2	$m+n+1$	2	mn	2	$\leq m2^n$	
Symmetric difference	$m+n$	3	$m+n$	3	mn	2	$\leq 2^{m+n}$	
Reversal	2^n	2	2^n	2	2^n	2	$n+1$	2
Star	2^n	2	2^n	2	$\frac{3}{4}2^n$	2	$n+1$	1
Left quotient	m	1	$m+1$	1	2^m-1	2	$m+1$	2
Right quotient	2^m	2	2^m+1	2	m	1	m	1

References

1. Brzozowski, J., Jirásková, G., Liu, B., Rajasekaran, A., Szykuła, M.: On the state complexity of the shuffle of regular languages. In: Câmpeanu, C., Manea, F., Shallit, J. (eds.) DCFS 2016. LNCS, vol. 9777, pp. 73–86. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-41114-9_6
2. Brzozowski, J.A., Leiss, E.L.: On equations for regular languages, finite automata, and sequential networks. Theoret. Comput. Sci. **10**, 19–35 (1980). https://doi.org/10.1016/0304-3975(80)90069-9
3. Čevorová, K.: Kleene star on unary regular languages. In: Jurgensen, H., Reis, R. (eds.) DCFS 2013. LNCS, vol. 8031, pp. 277–288. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39310-5_26
4. Chandra, A.K., Kozen, D., Stockmeyer, L.J.: Alternation. J. ACM **28**(1), 114–133 (1981). https://doi.org/10.1145/322234.322243
5. Fellah, A., Jürgensen, H., Yu, S.: Constructions for alternating finite automata. Int. J. Comput. Math. **35**(1–4), 117–132 (1990). https://doi.org/10.1080/00207169008803893
6. Holzer, M., Kutrib, M.: Nondeterministic descriptional complexity of regular languages. Int. J. Found. Comput. Sci. **14**(6), 1087–1102 (2003). https://doi.org/10.1142/S0129054103002199
7. Hopcroft, J.E., Ullman, J.D.: Introduction to Automata Theory, Languages and Computation. Addison-Wesley, Boston (1979)
8. Hospodár, M., Jirásková, G.: Concatenation on deterministic and alternating automata. In: Bordihn, H., Freund, R., Nagy, B., Vaszil, G. (eds.) NCMA 2016, vol. 321, pp. 179–194. Österreichische Computer Gesellschaft (2016). books@ocg.at
9. Jirásková, G.: State complexity of some operations on binary regular languages. Theoret. Comput. Sci. **330**(2), 287–298 (2005). https://doi.org/10.1016/j.tcs.2004.04.011
10. Jirásková, G.: Descriptional complexity of operations on alternating and Boolean automata. In: Hirsch, E.A., Karhumäki, J., Lepistö, A., Prilutskii, M. (eds.) CSR 2012. LNCS, vol. 7353, pp. 196–204. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30642-6_19
11. Jirásková, G., Šebej, J.: Reversal of binary regular languages. Theoret. Comput. Sci. **449**, 85–92 (2012). https://doi.org/10.1016/j.tcs.2012.05.008
12. Krajňáková, I., Jirásková, G.: Square on deterministic, alternating, and Boolean finite automata. In: Pighizzini, G., Câmpeanu, C. (eds.) DCFS 2017. LNCS, vol. 10316, pp. 214–225. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-60252-3_17
13. Leiss, E.L.: Succint representation of regular languages by Boolean automata. Theoret. Comput. Sci. **13**, 323–330 (1981). https://doi.org/10.1016/S0304-3975(81)80005-9
14. Maslov, A.N.: Estimates of the number of states of finite automata. Soviet Math. Doklady **11**(5), 1373–1375 (1970)
15. Nicaud, C.: Average state complexity of operations on unary automata. In: Kutyłowski, M., Pacholski, L., Wierzbicki, T. (eds.) MFCS 1999. LNCS, vol. 1672, pp. 231–240. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48340-3_21
16. Palmovský, M.: Kleene closure and state complexity. RAIRO - Theor. Inf. Appl. **50**(3), 251–261 (2016). https://doi.org/10.1051/ita/2016024
17. Sipser, M.: Introduction to the theory of computation. Cengage Learn (2012)

18. Yu, S.: Regular languages. In: Rozenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages. Volume 1: Word, Language, Grammar, pp. 41–110. Springer, Heidelberg (1997). https://doi.org/10.1007/978-3-642-59136-5_2
19. Yu, S., Zhuang, Q., Salomaa, K.: The state complexities of some basic operations on regular languages. Theoret. Comput. Sci. **125**(2), 315–328 (1994). https://doi.org/10.1016/0304-3975(92)00011-F