The Ranges of Accepting State Complexities of Languages Resulting From Some Operations

Michal Hospodár[1] and Markus Holzer[2](✉)

[1] Mathematical Institute, Slovak Academy of Sciences,
Grešákova 6, 040 01 Košice, Slovakia
hosmich@gmail.com

[2] Institut für Informatik, Universität Giessen,
Arndtstr. 2, 35392 Giessen, Germany
holzer@informatik.uni-giessen.de

Abstract. We examine the accepting state complexity, i.e., the minimal number of *accepting states* of deterministic finite automata (DFAs) for languages resulting from unary and binary operations on languages with accepting state complexity given as a parameter. This is continuation of the work of [J. Dassow: On the number of accepting states of finite automata, *J. Autom., Lang. Comb.*, 21, 2016]. We solve most of the open problems mentioned thereof. In particular, we consider the operations of intersection, symmetric difference, right and left quotients, reversal, and permutation (on finite languages), where we obtain precise ranges of the accepting state complexities.

1 Introduction

The descriptional complexity of regular languages, to be more precise the state complexity of deterministic and nondeterministic finite automata and regularity preserving operations thereof, is well understood. While the deterministic state complexity of a regular language can be read of from the minimal deterministic finite automaton (DFA) for the language in question, it is well-known that this is not the case for the nondeterministic state complexity. Moreover, it is folklore, that the deterministic and nondeterministic state complexity forms a strict infinite hierarchy w.r.t. the number of states. Yet another well known result is that for DFAs the number of accepting states is a host for an infinite strict hierarchy, while for nondeterministic finite automata (NFAs) two accepting states suffice, and if λ-transitions (spontaneous transitions) are allowed for NFAs even a single accepting state is enough to accept every regular language. But what else

M. Hospodár—Research supported by VEGA grant 2/0084/15 and grant APVV-15-0091. This work was conducted during a research visit at the Institut für Informatik, Universität Giessen, Germany, funded by the DAAD short-term grant ID 57314022.

C. Câmpeanu (Ed.): CIAA 2018, LNCS 10977, pp. 198–210, 2018.
https://doi.org/10.1007/978-3-319-94812-6_17

can be said about the number of accepting states for finite automata, in particular, when regularity preserving operations such as, e.g., Boolean operations, concatenation, Kleene star, etc., are applied to the finite state devices?

A partial answer to these questions was recently given in [3]. There the accepting state complexity of DFAs and NFAs was introduced and investigated in detail. To be more precise, the *(deterministic) accepting state complexity* of a regular language L is defined as the minimal number of accepting states needed for a DFA to accept L. Analogously one defines the *nondeterministic accepting state complexity* of a regular language. Similarly as for ordinary deterministic state complexity the deterministic *accepting state complexity* of a regular language can be determined from the minimal DFA for the language under consideration. On the other hand, the nondeterministic accepting state complexity is trivial as already mentioned above. The major contribution of [3] is the investigation of the deterministic accepting state complexity or for short the accepting state complexity w.r.t. the operations of complementation, union, concatenation, set difference, and Kleene star, which are summarized on the left in Table 1—a number within the range is *magic* if it cannot be produced by the operation from any K and L with the appropriate complexities. Hence, the quest to understand the accepting state complexity of operations can be seen as a variant of the magic number problem—see, e.g., [5,8,10], but now for the descriptional complexity measure accepting states instead of ordinary states.

Table 1. Results obtained in [3] (left) and the results of this paper (right). It is assumed that K and L have accepting state complexity m and n, respectively, for $m, n \geq 1$. Then the range indicates the obtainable accepting state complexities of the operation under consideration and the status of the magic number problem refers to whether there are magic numbers in the given range or not.

Op.	Range	Magic num.	$\|\Sigma\|$	Op.	Range	Magic num.	$\|\Sigma\|$
$\Sigma^* \setminus L$	$\mathbb{N} \cup \{\, 0 \mid n = 1 \,\}$	no	1	$K \cap L$	$[0, mn]$	no	2
$K \cup L$	$\mathbb{N}$	no	1	$K \oplus L$	$\{0\} \cup \mathbb{N}$	no	1
KL	$\mathbb{N}$	no	1	KL^{-1}	$\{0\} \cup \mathbb{N}$	no	1
$K \setminus L$	$\{0\} \cup \mathbb{N}$	no	1	$L^{-1}K$	$\{0\} \cup \mathbb{N}$	no	1
L^*	$\mathbb{N}$	no	1	L^R	$\mathbb{N}$	no	2
$K \cap L$	$[0, mn]$	?		$\mathrm{per}(L)$	$\mathbb{N} \setminus \{\, 1 \mid n \geq 2 \,\}$	no	2

This is the starting point of our investigation. We study the accepting state complexity of the operations intersection, symmetric difference, right and left quotients, reversal, and permutation. The latter operation is only considered on finite languages, since regular languages are not closed under permutation. The obtained results are summarized on the right of Table 1. We solve most open problems from [3]. It is worth mentioning that intersection has an accepting state complexity bounded from above and no magic numbers within this interval.

2 Preliminaries

We recall some definitions on finite automata as contained in [6]. Let Σ^* denote the set of all words over the finite alphabet Σ. The *empty word* is denoted by λ. Further, we denote the set $\{i, i+1, \ldots, j\}$ by $[i, j]$, if i and j are integers.

A *nondeterministic finite automaton* (NNFA) is a 5-tuple $A = (Q, \Sigma, \delta, I, F)$, where Q is a finite set of states, Σ is a finite nonempty alphabet, $\delta \colon Q \times \Sigma \to 2^Q$ 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 accepting (or final) states. We say that (p, a, q) is a *transition* in A if $q \in \delta(p, a)$. If (p, a, q) is a transition in A, then we say that the state q has an *ingoing transition*, and the state p has an *outgoing transition*. We sometimes write $p \xrightarrow{w} q$, if $q \in \delta(p, w)$. The *language accepted by* A is the set $L(A) = \{\, w \in \Sigma^* \mid \delta(I, w) \cap F \neq \emptyset \,\}$. If $|I| \geq 2$, we say that A is a *nondeterministic finite automaton with nondeterministic choice of initial state* (so we use the abbreviation NNFA, cf. [15]). Otherwise, if $|I| = 1$, we say that A is a *nondeterministic finite automaton* (NFA). In this case we simple $A = (Q, \Sigma, \delta, s, F)$ instead of $A = (Q, \Sigma, \delta, \{s\}, F)$. Moreover, an NFA A is a (partial) *deterministic finite automaton* (DFA), if $|\delta(q, a)| \leq 1$, for each q in Q and each a in Σ, and it is a *complete* DFA, if $|\delta(q, a)| = 1$, for each q in Q and each a in Σ.

Every NNFA $A = (Q, \Sigma, \delta, I, F)$ can be converted to an equivalent complete DFA $\mathcal{D}(A) = (2^Q, \Sigma, \delta, I, \{\, S \in 2^Q \mid S \cap F \neq \emptyset \,\})$, where $\delta(S, a) = \bigcup_{q \in S} \delta(q, a)$, for $S \in 2^Q$ and $a \in \Sigma$. We call the DFA $\mathcal{D}(A)$ the *subset automaton* of A.

The *state complexity of a regular language* L, referred to as $\mathrm{sc}(L)$, is the smallest number of states in any complete DFA accepting L. The *state complexity of a regular operation* is the number of states that are sufficient and necessary in the worst case for a DFA to accept the language resulting from the operation, considered as a function of the number of states of DFAs for the given operands. Similarly we define the *accepting state complexity* of a language L by

$$\mathrm{asc}(L) = \min\{\, n \mid L \text{ is accepted by a DFA with } n \text{ accepting states} \,\}.$$

An automaton is *minimal* (a-minimal, respectively) if it admits no smaller equivalent automaton w.r.t. the number of states (accepting states, respectively). For DFAs both properties can be easily verified. Minimality can be shown if all states are reachable from the initial state and all states are pairwise inequivalent. For a-minimality the following result shown in [3, Theorem 1] applies.

Theorem 1. *Let* L *be a language accepted by a minimal* DFA A. *Then the number of accepting states of* A *is equal to* $\mathrm{asc}(L)$.

Note that a-minimality can be shown if all states are reachable from the initial state and all *accepting* states are pairwise inequivalent. In fact, we do not need to prove distinguishability of all (including rejecting) states.

In order to characterize the behaviour of complexities under operations we introduce the following notation: for $c \in \{\mathrm{sc}, \mathrm{asc}\}$, a k-ary regularity preserving operation $\circ$ on languages, and natural numbers $n_1, n_2, \ldots, n_k$, we define

$$g_\circ^c(n_1, n_2, \ldots, n_k)$$

as the set of all integers α such that there are k regular languages $L_1, L_2, \ldots, L_k$ with $c(L_i) = n_i$, for $1 \leq i \leq k$, and $c(\circ(L_1, L_2, \ldots, L_k)) = \alpha$. In case we only consider unary (finite, respectively) languages $L_1, L_2, \ldots, L_k$ we write $g_\circ^{c,u}$ ($g_\circ^{c,f}$, respectively) instead. Let $I_\circ^c$ be the smallest integer interval containing all elements from the set $g_\circ^c(n_1, n_2, \ldots, n_k)$. Then any element from $I_\circ^c \setminus g_\circ^c(n_1, n_2, \ldots, n_k)$ is said to be a *magic number* for the operation $\circ$ with respect to the complexities $n_1, n_2, \ldots, n_k$. This notion was introduced in [8,9].

The *nondeterministic accepting state complexity* of a language L, denoted by $\mathrm{nasc}(L)$, refers to the minimal number of accepting states in any NFA for L. It was shown in [3] that for every nonempty regular language L we have $\mathrm{nasc}(L) = 1$, if $\lambda \notin L$, but $\mathrm{nasc}(L) \leq 2$, if $\lambda \in L$. Thus, the nondeterministic accepting state complexity is not too interesting. Nevertheless, it was left open to give a sufficient and necessary condition for a language L such that $\mathrm{nasc}(L) = 1$ and $\lambda \in L$. This problem was solved in [11].

Lemma 2. *A language L satisfies $\lambda \in L$ and* $\mathrm{nasc}(L) = 1$ *if and only if $L = L^*$.*

Proof. If $\lambda \in L$ and $\mathrm{nasc}(L) = 1$, then there is an NFA for L in which the single accepting state is the initial state. Therefore $L = L^*$.

Conversely, let $A = (Q, \Sigma, \delta, s, F)$ be an NFA accepting the set L. If $L = L^*$, then $\lambda \in L$, so the initial state s of A is accepting. For every accepting state q_f in $F \setminus \{s\}$ and every transition (q, a, q_f) we add the transition (q, a, s) to A and make the state q_f rejecting. Since $L = L^*$, the resulting automaton, which has exactly one accepting state, accepts L. It follows that $\mathrm{nasc}(L) = 1$. □

3 Results

We investigate the accepting state complexity of various regularity preserving language operations such as, e.g., intersection, symmetric difference, right and left quotients, reversal, and permutation on finite languages. We start with the accepting state complexity of intersection solving an open problem stated in [3].

3.1 Intersection

For two DFAs $A = (Q_A, \Sigma, \delta_A, s_A, F_A)$ and $B = (Q_B, \Sigma, \delta_B, s_B, F_B)$ we apply the standard cross-product construction in order to construct an automaton for the intersection of $L(A)$ and $L(B)$. Thus, define $C = (Q_C, \Sigma, \delta_C, s_C, F_C)$ with $Q_C = Q_A \times Q_B$, $s_C = (q_A, q_B)$, and $F_C = F_A \times F_B$. The transition function is set to $\delta_C((p, q), a) = (\delta_A(p, a), \delta_B(q, a))$. Thus, we have $L(C) = L(A) \cap L(B)$. If A is an m-state and B an n-state DFA then the above construction results in an

mn-state DFA C. In [16] it was shown that this upper bound is necessary in the worst case, that is, it can be reached by two appropriately chosen minimal DFAs with m and n states, respectively. Moreover, in [7] it was shown that there are no magic numbers for intersection on a binary alphabet. This is a direct consequence of the theorem that there are no magic numbers for union, De Morgan's law, and the fact the complementation preserves the state complexity. Thus, for every α with $1 \leq \alpha \leq mn$ there are minimal m-state and n-state DFAs such that the intersection of the languages described by these automata requires a minimal DFA with exactly α states.

Now let us turn our attention to the accepting state complexity of intersection. The next theorem solves an open problem stated in [3].

Theorem 3. *We have* $g_{\cap}^{\text{asc}}(m,n) = g_{\cap}^{\text{asc}}(n,m) = [0, mn]$.

Proof. Since intersection is commutative we have $g_{\cap}^{\text{asc}}(m,n) = g_{\cap}^{\text{asc}}(n,m)$. Now let $0 \leq \alpha \leq mn$. We are going to describe minimal DFAs A and B with m and n accepting states, respectively, such that $\text{asc}(L(A) \cap L(B)) = \alpha$. Notice that α can be expressed as $\alpha = kn + \ell$, for some integers k and ℓ with $0 \leq k \leq m$ and $0 \leq \ell \leq n-1$.

Define the DFA $A = ([1, m+1], \{a, b\}, \delta_A, m+1, [1, m])$, where

$$\delta_A(i,a) = i-1, \text{ if } 2 \leq i \leq m+1; \qquad \delta_A(i,b) = \begin{cases} i, & \text{if } 1 \leq i \leq m; \\ k+1, & \text{if } i = m+1. \end{cases}$$

Next, define the DFA $B = ([0, n+1], \{a, b\}, \delta_B, n+1, [1, n])$, where

$$\delta_B(j,a) = n, \text{ if } j = 0; \qquad \delta_B(j,b) = \begin{cases} j-1, & \text{if } 1 \leq j \leq n; \\ \ell, & \text{if } j = n+1. \end{cases}$$

The DFAs A and B are depicted in Fig. 1. It is easy to see that both DFAs are minimal.

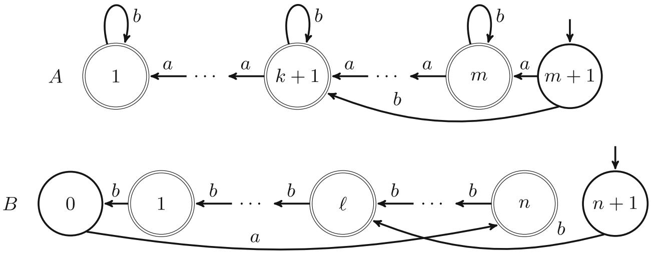

Fig. 1. Let α satisfy $0 \leq \alpha \leq mn$. The witness DFAs A (top) and B (bottom) for intersection with $\alpha = kn + \ell$, for $0 \leq k \leq m$ and $0 \leq \ell \leq n$.

We construct the automaton C of A and B according to the previously given construction. The product automaton C has the following transitions:

1. $\rightarrow (m+1, n+1) \xrightarrow{b} (k+1, \ell) \xrightarrow{b} (k+1, \ell-1) \xrightarrow{b} \ldots \xrightarrow{b} (k+1, 1) \xrightarrow{b} (k+1, 0)$,
2. $(k+1, 0) \xrightarrow{a} (k, n) \xrightarrow{b} (k, n-1) \xrightarrow{b} \ldots \xrightarrow{b} (k, 1) \xrightarrow{b} (k, 0)$,
3. $(k, 0) \xrightarrow{a} (k-1, n) \xrightarrow{b} (k-1, n-1) \xrightarrow{b} \cdots \xrightarrow{b} (k-1, 1) \xrightarrow{b} (k-1, 0)$, etc., and
4. $(2, 0) \xrightarrow{a} (1, n) \xrightarrow{b} (1, n-1) \xrightarrow{b} \ldots \xrightarrow{b} (1, 1) \xrightarrow{b} (1, 0)$.

No other transitions are present in C. It follows that $L(C)$ is a finite language with the longest word $b^{\ell+1}(ab^n)^{k-1}ab^{n-1}$. Hence every NFA for the language $L(C)$ has at least $k(n+1)+\ell+1$ states. Thus C with the state $(1,0)$ removed is a minimal NFA for $L(C) = L(A) \cap L(B)$. Next, since C is a DFA it is a minimal DFA. So every state pair is distinguishable. Note that the states (i,j), for $1 \leq i \leq k$ and $1 \leq j \leq n$, and $(k+1, j)$, for $1 \leq j \leq \ell$, are reachable and accepting in C. It follows that we have $kn+\ell$ reachable and pairwise distinguishable accepting states. Thus $\mathrm{asc}(L(A) \cap L(B)) = kn + \ell = \alpha$, and the theorem follows. □

3.2 Symmetric Difference

The symmetric difference ($\oplus$) of two languages accepted by finite automata can also be obtained by a product construction, similar as in the case of intersection. The only difference to the construction used for intersection is the definition of the set of accepting states, which in case of symmetric difference is set to $F_C = F_A \times (Q_B \setminus F_B) \cup (Q_A \setminus F_A) \times F_B$, where the notation is that for intersection used in Subsect. 3.1. Thus, for the ordinary state complexity the upper bound is mn, which was shown to be tight in [17]. To our knowledge the magic number problem for state complexity of the symmetric difference operation was not investigated so far. For the accepting state complexity we find the following situation, where we utilize the fact that for unary finite and unary co-finite languages it is very easy to determine the number of accepting states, from a description of the language in question. For instance, the unary language $L = \{\, a^i \mid 2 \leq i \leq 5\} \cup \{\, a^j \mid j \geq 7\,\}$ is accepted by a minimal DFA with $(5-2)+1+1 = 5$ accepting states. For the structure of (minimal) unary DFAs in general we refer to [2].

Lemma 4. *Let $m, n \geq 1$ and $m \leq n$. Then for every α with $\alpha \geq 1$ there are minimal unary* DFA*s A and B with m and n accepting states, respectively, such that the minimal* DFA *for $L(A) \oplus L(B)$ has α accepting states.*

Proof. Define the unary languages $K = \{\, a^i \mid 0 \leq i \leq m-2 \text{ or } i \geq m\,\}$ and $L = \{\, a^i \mid 0 \leq i \leq m-2 \text{ or } m \leq i \leq n-1 \text{ or } i \geq n+\alpha\,\}$. Let A and B be minimal DFAs for K and L, respectively. Then A and B have m and n accepting states, respectively. Moreover, $L(A) \oplus L(B) = \{\, a^i \mid n \leq i \leq n+\alpha-1\,\}$, which is accepted by a minimal DFA with α accepting states. □

Now we are ready to describe the behaviour of the accepting state complexity measure w.r.t. the symmetric difference operation.

Theorem 5. *We have*

$$g_{\oplus}^{\mathrm{asc},u}(m,n)=g_{\oplus}^{\mathrm{asc}}(m,n)=g_{\oplus}^{\mathrm{asc}}(n,m)=\begin{cases}\{n\} & \textit{if}\, m=0;\\ \{m\} & \textit{if}\, n=0;\\ \{0\}\cup\mathbb{N} & \textit{if}\, m,n\geq 1\,\textit{and}\, m=n;\\ \mathbb{N} & \textit{otherwise.}\end{cases}$$

Proof. The symmetric difference of two languages is commutative. Therefore $g_{\oplus}^{\mathrm{asc}}(m,n)=g_{\oplus}^{\mathrm{asc}}(n,m)$. The only language with accepting state complexity 0 is the empty language $\emptyset$. For nonempty languages Lemma 4 applies. Since we have $\emptyset\oplus L=L$, $K\oplus\emptyset=K$, and $K\oplus L=\emptyset$ if and only if $K=L$, the first three cases of $g_{\oplus}^{\mathrm{asc}}$ are covered. Thus, all natural numbers can be obtained as the number of accepting states of a DFA accepting the symmetric difference of DFAs A and B with m and n accepting states, respectively; notice that $m\neq n$ implies $K\neq L$. Additionally, in case $m=n$ one can also obtain the value 0, since in this case we can force both languages K and L to be the same, which gives $K\oplus L=\emptyset$. Finally, $g_{\oplus}^{\mathrm{asc}}(m,n)=g_{\oplus}^{\mathrm{asc},u}(m,n)$ since all our witnesses are unary languages. □

3.3 Right and Left Quotients

The *right quotient* of a language K by a language L is defined as follows:

$$KL^{-1}=\{\,w\mid \text{there is a } x\in L \text{ such that } wx\in K\,\}.$$

The DFA accepting KL^{-1} is the same as the DFA accepting K except that the set of accepting states is different. To be more precise, let $A=(Q,\Sigma,\delta,s,F)$ be the DFA accepting K, then $B=(Q,\Sigma,\delta,s,\{\,q\mid\exists\, x\in L:\delta(q,x)\in F\,\})$ accepts the language KL^{-1}. Thus, for an m-state DFA the upper bound for the state complexity of the right quotient w.r.t. any language is m, which is known to be tight [17]. Similarly one defines the *left quotient* of K by L as

$$L^{-1}K=\{\,w\mid \text{there is a } x\in L \text{ such that } xw\in K\,\}.$$

It was proven that for an m-state DFA language K, the state complexity of the left quotient of K by any language L is at most 2^m-1. Again, this bound is tight [17]. Note, when considering unary languages K and L, the right and left quotient coincide, i.e., $KL^{-1}=L^{-1}K$. Thus, in this case, the state complexity is bounded by the state complexity of K. To our knowledge the magic number problem for state complexity of the quotient operations was not investigated so far. Next we consider the magic number problem for accepting state complexity of the quotient operations.

Lemma 6. *Let $m,n\geq 1$. Then for every α with $\alpha\geq 0$ there are minimal unary* DFA*s A and B with m and n accepting states, respectively, such that the minimal* DFA *for $L(A)L(B)^{-1}$ has α accepting states.*

Proof. We consider two cases:

1. Let $\alpha < n$. Define the languages $K = \{\, a^i \mid 0 \leq i \leq m-2 \text{ or } i = m+\alpha \,\}$ and $L = \{\, a^i \mid m+1 \leq i \leq m+n \,\}$. The language K (L, respectively) is accepted by a minimal DFA with m (n, respectively) accepting states. Next $KL^{-1} = \{\, a^i \mid 0 \leq i \leq \alpha - 1 \,\}$, whose minimal DFA has α accepting states. Observe, that this case also covers $\alpha = 0$, where KL^{-1} becomes empty.
2. Now let $\alpha \geq n$. Let K be the same language as above and define the set $L = \{\, a^i \mid m \leq i \leq m+n-2 \text{ or } i \geq m+n \,\}$. The language K (L, respectively) is accepted by a minimal DFA with m (n, respectively) accepting states. Next $KL^{-1} = \{\, a^i \mid 0 \leq i \leq \alpha - n \text{ or } \alpha - n + 2 \leq i \leq \alpha \,\}$, whose minimal DFA has α accepting states. □

In the next theorem we use an alternative notation for the quotients, namely $K/L := KL^{-1}$ for the right quotient and $L\backslash K := L^{-1}K$ for the left quotient.

Theorem 7. *We have* $g_{/}^{\mathrm{asc},u}(m,n) = g_{/}^{\mathrm{asc}}(m,n)$ *and*

$$g_{/}^{\mathrm{asc}}(m,n) = \begin{cases} \{0\} & \textit{if} m = 0 \textit{ or } n = 0; \\ \{0\} \cup \mathbb{N} & \textit{otherwise.} \end{cases}$$

Next, we have $g_{\backslash}^{\mathrm{asc}}(m,n) = g_{/}^{\mathrm{asc}}(m,n)$ *and* $g_{\backslash}^{\mathrm{asc},u}(m,n) = g_{/}^{\mathrm{asc},u}(m,n)$. □

3.4 Reversal

As usual, the *reverse of a word* over Σ is defined by $\lambda^R = \lambda$ and $(va)^R = av^R$, for every a in Σ and v in Σ^*. The *reverse of a language* L is defined as $L^R = \{\, w^R \mid w \in L \,\}$. In order to obtain an NNFA accepting the reverse of a language L accepted by a DFA $A = (Q, \Sigma, \delta, s, F)$ one reverses all transitions and swaps the role of initial and accepting states. This results in an NNFA that accepts the language L^R. More formally, this automaton can be described as $A^R = (Q, \Sigma, \delta^R, F, \{s\})$, where $\delta^R(p,a) = \{\, q \in Q \mid \delta(q,a) = p \,\}$. Finally, we obtain the DFA $\mathcal{D}(A^R)$ for the language L^R, which provides the upper bound 2^n on the state complexity of the reversal operation on complete DFAs. In [13] it was shown that this bound is tight for languages over an alphabet of at least two letters. This alphabet size is optimal since the reverse of every unary language is the same language, hence n is a tight upper bound for the ordinary state complexity of the reversal operation. Moreover, every value from $\log n$ to 2^n can be obtained as the state complexity of L^R if the state complexity of L is n [14].

Before we consider the accepting state complexity of the reversal operation we take a closer look on the automaton $\mathcal{D}(A^R)$. Observe, that the state s is the single accepting state of the NNFA A^R. Therefore the accepting subsets of the corresponding subset automaton $\mathcal{D}(A^R)$ are those containing the state s. Moreover, if A is a DFA without unreachable states, then the subset automaton $\mathcal{D}(A^R)$ does not have equivalent states [12, Proposition 3]. Now we are ready to consider accepting state complexity of reversal in general.

Lemma 8. *Let $n \geq 1$. Then for every α with $\alpha \geq 1$ there exists a minimal binary* DFA *A with n accepting states such that the minimal* DFA *for $L(A^R)$ has α accepting states.*

Proof. Let $A = ([1, \alpha + n], \{a, b\}, \delta, 1, F)$, where $F = [\alpha + 1, \alpha + n]$, and

$$\delta(i, a) = \begin{cases} i & \text{if } i = 1 \text{ or } i = \alpha + 1; \\ i - 1 & \text{otherwise,} \end{cases} \qquad \delta(i, b) = \begin{cases} \alpha + n & \text{if } i = 1; \\ \alpha & \text{if } i = \alpha + 1. \end{cases}$$

The DFA A is shown in Fig. 2. Two rejecting states are distinguished by a word in a^*b and two accepting states by a word in $a^*ba^{\alpha-1}b$. Hence A is minimal.

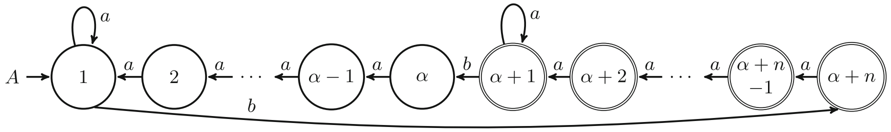

Fig. 2. The witness DFA A for the reversal operation with $n, \alpha \geq 1$.

We construct the NNFA $A^R = ([1, \alpha + n], \{a, b\}, \delta^R, F, \{1\})$ from the DFA A by reversing all the transitions, and by swapping the roles of the initial and accepting states. The subset automaton $\mathcal{D}(A^R)$ has the initial state F and the following transitions:

1. $\rightarrow F \xrightarrow{a} F \xrightarrow{b} \{1\} \xrightarrow{a} [1, 2] \xrightarrow{a} [1, 3] \xrightarrow{a} \cdots \xrightarrow{a} [1, \alpha] \xrightarrow{a} [1, \alpha] \xrightarrow{b} \{\alpha + 1\}$ and
2. $\{\alpha + 1\} \xrightarrow{a} [\alpha + 1, \alpha + 2] \xrightarrow{a} [\alpha + 1, \alpha + 3] \xrightarrow{a} \cdots \xrightarrow{a} [\alpha + 1, \alpha + n - 1] \xrightarrow{a} F$.

Since every other transition from these reachable states goes to the empty set, no more states are reachable. Since only the subsets containing 1 are accepting, there are α reachable accepting subsets. By [12, Proposition 3], the subset automaton $\mathcal{D}(A^R)$ does not have equivalent states, and the theorem follows. □

Taking into account that the only language with accepting state complexity 0 is the empty language $\emptyset$, and for nonempty languages Lemma 8 applies, we obtain the next result. Moreover, since the reverse of a unary language is the same language, we immediately get the result on the accepting state complexity of reversal for unary regular languages, too.

Theorem 9. *We have*

$$g_R^{\text{asc}}(n) = \begin{cases} \{0\} & \textit{if } n = 0; \\ \mathbb{N} & \textit{otherwise.} \end{cases}$$

For unary regular languages, we have $g_R^{\text{asc},u}(n) = \{n\}$, *if* $n \geq 0$. □

3.5 Permutation on Finite Languages

The *permutation* of a language L is defined as $\text{per}(L) = \bigcup_{w \in L} \text{per}(w)$, where $\text{per}(w) = \{\, u \in \Sigma^* \mid \psi(u) = \psi(w) \,\}$ with $\psi(v) = (|v|_{a_1}, |v|_{a_2}, \ldots, |v|_{a_k})$, the

Parikh vector of a word v over the alphabet $\Sigma = \{a_1, a_2, \ldots, a_k\}$. Here $|v|_a$ refers to the number of occurrences of the letter a in v. It is known that the permutation operation is not regular on infinite languages. For example, $\mathrm{per}(\{ab\}^*) = \{\, w \in \{a,b\}^* \mid |w|_a = |w|_b \,\}$ is *not* regular. On the other hand, permutation of a finite language is always finite, and every finite language is regular. So permutation is a regular operation on finite languages. Moreover, note that every unary language is a permutation of itself, thus one may consider the ordinary state as well as the accepting state complexity of permutation on *binary* finite languages. Ordinary deterministic state complexity was considered in [1], where an upper bound of $\frac{n^2-n+2}{2}$ states for the permutation of a finite binary language with state complexity n was shown. This was slightly improved for permutations of chain DFAs where a matching upper and lower bound was obtained. To our knowledge the magic number problem for state complexity of permutation on (binary) finite languages was not considered so far. For the accepting state complexity we can prove the following three lemmata:

Lemma 10. *Let $n \geq 1$. Then for every α with $\alpha \geq n$ there exists a minimal binary* DFA *A with n accepting states such that the minimal* DFA *for* $\mathrm{per}(L(A))$ *has α accepting states.*

Proof. Define the finite language $L = \{\, b^i a b^j \mid 0 \leq i \leq \alpha - n \text{ and } 0 \leq j \leq n-1 \,\}$. Since $L = \bigcup_{0 \leq j \leq n-1} [ab^j]$, where $[ab^j]$ is the Myhill-Nerode equivalence class with $[ab^j] = \{\, b^i a b^j \mid 0 \leq i \leq \alpha - n \,\}$, it is accepted by a minimal DFA with n accepting states. Observe, that every word w in L satisfies $|w|_a = 1$ and $0 \leq |w|_b \leq \alpha - 1$. Thus, $\mathrm{per}(L) = \{\, w \in \{a,b\}^* \mid |w|_a = 1 \text{ and } 0 \leq |w|_b \leq \alpha - 1 \,\}$. Hence, $\mathrm{per}(L) = \bigcup_{0 \leq i \leq \alpha-1} [ab^i]$, where $[ab^i]$ is now the Myhill-Nerode equivalence class $[ab^i] = \{\, w \in \{a,b\}^* \mid |w|_a = 1 \text{ and } |w|_b = i \,\}$. Therefore, we deduce that $\mathrm{per}(L)$ has accepting state complexity α. □

The next lemma follows from [4, Lemma 1].

Lemma 11. *Let $n \geq 2$. Let L be a finite language accepted by a minimal* DFA *with n accepting states. Then the minimal* DFA *for* $\mathrm{per}(L)$ *has at least* 2 *accepting states.* □

The magic status of numbers from 2 to n is considered next.

Lemma 12. *Let $n \geq 2$. Then for every α with $2 \leq \alpha \leq n$ there exists a minimal binary* DFA *A with n accepting states such that the minimal* DFA *for* $\mathrm{per}(L(A))$ *has α accepting states.*

Proof. We prove a slightly stronger statement, namely: let $m \geq 1$. Then for every α with $\alpha \geq 2$ there is a minimal binary DFA A with $2^m + (\alpha - 1)$ accepting states such that the minimal DFA for $\mathrm{per}(L(A))$ has α accepting states. The idea for the construction is as follows: for a word $w \in \{a,b\}^m$ let x_w refer to the length m word $b^{m-|w|_a} a^{m-|w|_b}$. Then define the finite language

$$L = \{\, w x_w \in \{a,b\}^* \mid |w| = m \,\} \cup \{\, w x_w w^R y \in \{a,b\}^* \mid |w| = m \text{ and } 0 \leq |y| \leq \alpha - 2 \,\}.$$

By construction every word of the form wx_w, for $w \in \{a,b\}^m$, has the Parikh vector (m,m). Moreover, the Parikh vector of every word of the form wx_ww^R, for $w \in \{a,b\}^m$, lies in the set $\{\,(m+i, 2m-i) \mid 0 \leq i \leq m\,\}$. By considering the Myhill-Nerode equivalence classes for the words in L one deduces that the accepting state complexity of L is $2^m + (\alpha - 1)$.

The automaton B accepting $\mathrm{per}(L)$ is constructed according to [1, Lemma 3.1]. Thus, the DFA B has a grid like structure (with a truncated lower right) where the b-transitions connect neighboring columns and the a-transitions neighboring rows and every state can be identified with a Parikh vector. A schematic drawing is given on the left of Fig. 3. The states in B that correspond to a Parikh vector of a word in L are marked accepting. Since every word wx_w, for $w \in \{a,b\}^m$, has the Parikh vector (m,m), the corresponding state is marked accepting—see the accepting state in the middle of the schematic drawing on the left of Fig. 3. The words of the form wx_ww^R, for $w \in \{a,b\}^m$, which Parikh vector lies in the set $\{\,(m+i, 2m-i) \mid 0 \leq i \leq m\,\}$ induce the topmost anti-diagonal of accepting states. This anti-diagonal is followed by $\alpha - 2$ further anti-diagonals of accepting states, since every word wx_ww^R can be extended by any word of length at most $\alpha - 2$. Again, see the left of Fig. 3. A close inspection reveals that this automaton is not minimal, because all states in a fixed anti-diagonal are equivalent. A schematic drawing of the minimal DFA accepting the permutation of the finite language L is shown on the right of Fig. 3. The tedious details of the construction are left to the reader.

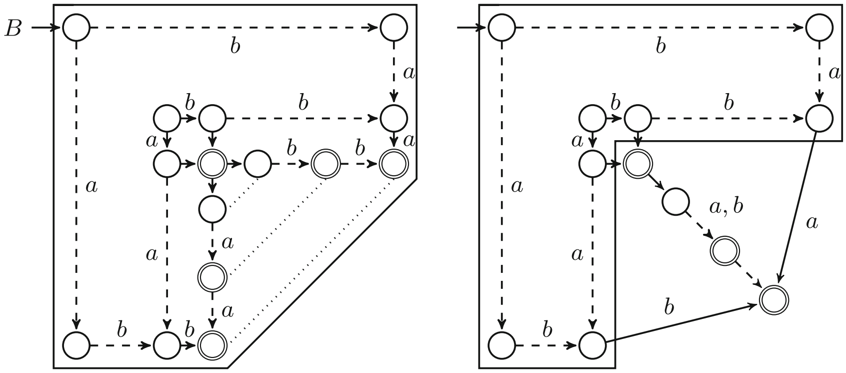

Fig. 3. A schematic drawing of the grid like DFA B (left) accepting $\mathrm{per}(L(A))$ and its minimal DFA (right) obtained from B by identifying accepting states that are connect by dotted lines.

In order to decrease the accepting state complexity of L one removes all words with prefix wx_w, for some words $w \in \{a,b\}^m$. Let L' refer to the resulting language. In order to keep the construction working as described above, one must ensure that all accepting states in the topmost anti-diagonal can be reached. This requirement is fulfilled if the Parikh vectors of all words wx_ww^R with $wx_w \in L'$

form the set $\{\,(m+i, 2m-i) \mid 0 \leq i \leq m\,\}$, which can always be achieved. Thus, at least $m+1$ words of the form wx_w, for $w \in \{a,b\}^*$, must belong to L'. Finally, this allows us to set the accepting state complexity of L' to n by choosing the parameter m appropriately, which proves the original statement. □

Taking into account Lemmata 10, 11, and 12, we get the following result.

Theorem 13. *We have*

$$g_{\text{per}}^{\text{asc}}(n) = g_{\text{per}}^{\text{asc},f}(n) = \begin{cases} \{0\} & \text{if } n = 0; \\ \mathbb{N} & \text{if } n = 1; \\ \mathbb{N} \setminus \{1\} & \text{if } n \geq 2. \end{cases}$$

For unary regular languages, we have $g_{\text{per}}^{\text{asc},u}(n) = \{n\}$ *if* $n \geq 0$. □

References

1. Cho, D.-J., Goč, D., Han, Y.-S., Ko, S.-K., Palioudakis, A., Salomaa, K.: State complexity of permutation on finite languages over a binary alphabet. Theoret. Comput. Sci. **682**, 67–78 (2017)
2. Chrobak, M.: Finite automata and unary languages. Theoret. Comput. Sci. **47**, 149–158 (1986)
3. Dassow, J.: On the number of accepting states of finite automata. J. Autom. Lang. Comb. **21**(1–2), 55–67 (2016)
4. Dassow, J.: Descriptional complexity and operations – two non-classical cases. In: Pighizzini, G., Câmpeanu, C. (eds.) DCFS 2017. LNCS, vol. 10316, pp. 33–44. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-60252-3_3
5. Geffert, V.: Magic numbers in the state hierarchy of finite automata. Inf. Comput. **205**(11), 1652–1670 (2007)
6. Harrison, M.A.: Introduction to Formal Language Theory. Addison-Wesley, Boston (1978)
7. Hricko, M., Jirásková, G., Szabari, A.: Union and intersection of regular languages and descriptional complexity. In: DCFS 2005, pp. 170–181 (2005)
8. Iwama, K., Kambayashi, Y., Takaki, K.: Tight bounds on the number of states of DFAs that are equivalent to n-state NFAs. Theoret. Comput. Sci. **237**(1–2), 485–494 (2000)
9. Iwama, K., Matsuura, A., Paterson, M.: A family of NFAs which need $2^n - \alpha$ deterministic states. Theoret. Comput. Sci. **301**(1–3), 451–462 (2003)
10. Jirásková, G.: Magic numbers and ternary alphabet. Internat. J. Found. Comput. Sci. **22**(2), 331–344 (2011)
11. Jirásková, G.: Personal communication (2017)
12. Jirásková, G., Šebej, J.: Reversal of binary regular languages. Theoret. Comput. Sci. **449**, 85–92 (2012)
13. Leiss, E.: Succinct representation of regular languages by Boolean automata. Theoret. Comput. Sci. **13**, 323–330 (1981)
14. Šebej, J.: Reversal on regular languages and descriptional complexity. In: Jurgensen, H., Reis, R. (eds.) DCFS 2013. LNCS, vol. 8031, pp. 265–276. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39310-5_25

15. Yu, S.: Regular languages. In: Rozenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages, vol. 1, pp. 41–110. Springer, Heidelberg (1997). https://doi.org/10.1007/978-3-642-59136-5
16. Yu, S., Zhuang, Q.: On the state complexity of intersection of regular languages. SIGACT News **22**(3), 52–54 (1991)
17. Yu, S., Zhuang, Q., Salomaa, K.: The state complexity of some basic operations on regular languages. Theoret. Comput. Sci. **125**, 315–328 (1994)