On the Accepting State Complexity of Operations on Permutation Automata

Christian Rauch Markus Holzer
Institut für Informatik, Universität Giessen, Arndstr. 2, 35392 Giessen, Germany
christian.rauch@informatik.uni-giessen.de holzer@informatik.uni-giessen.de

We investigate the accepting state complexity of deterministic finite automata for regular languages obtained by applying one of the following operations to languages accepted by permutation automata: union, quotient, complement, difference, intersection, Kleene star, Kleene plus, and reversal. The paper thus joins the study of accepting state complexity of regularity preserving language operations which was initiated by the work [J. Dassow: On the number of accepting states of finite automata, J. Autom., Lang. Comb., 21, 2016]. We show that for almost all of the operations, except for reversal and quotient, there is no difference in the accepting state complexity for permutation automata compared to deterministic finite automata in general. For both reversal and quotient we prove that certain accepting state complexities cannot be obtained; these number are called "magic" in the literature. Moreover, we solve the left open accepting state complexity problem for the intersection of unary languages accepted by permutation automata and deterministic finite automata in general.

1 Introduction

The state complexity of a regular language is a classical well-understood descriptional complexity measure of finite state systems, that is defined to be the number of states of the smallest, either deterministic or nondeterministic, finite automaton that recognizes it. It has been studied from different perspectives in the literature like, for instance, (i) for regular languages in general and for certain sub-families, (ii) for converting nondeterministic finite automata to equivalent deterministic finite automata, and (iii) for operations, called the operational complexity, on regular languages in general and sub-families thereof. For a brief survey on the subject we refer to, e.g., [3].

Recently, the accepting state complexity of a regular language was introduced in [2]. It is defined to be the minimal number of accepting states needed for a finite state device, either deterministic or nondeterministic, that accepts it. While the accepting state complexity forms a strict hierarchy of language classes for deterministic finite automata, it collapses for nondeterministic state devices, since every regular language not containing the empty word is accepted by a nondeterministic finite automaton with a single final state. If the empty word belongs to the language, the nondeterministic accepting state complexity is at most two. Thus, the conversion from nondeterministic to equivalent deterministic finite automata can produce unbounded deterministic accepting state complexity for a regular language. Moreover, the operational accepting state complexity was studied in [6]. The obtained results on the accepting state complexity prove that this measure is significantly different to the original state complexity. What is missing for the accepting state complexity is a study for certain sub-families of the regular languages in order to better understand the intrinsic behaviour of this measure.

We close this gap by studying the operational accepting state complexity for the class of permutation automata (PFAs) which accept the so called p-regular languages, also named pure-group languages. This language family is of particular interest from an algebraic point of view since their syntactic monoid

H. Bordihn, G. Horváth, G. Vaszil (Eds.): 12th International Workshop on Non-Classical Models of Automata and Applications (NCMA 2022)
EPTCS 367, 2022, pp. 177–189, doi:10.4204/EPTCS.367.12

induces a group. Additionally permutation automata together with permutation-reset automata play a key role in the decomposition of deterministic finite automata (DFAs), see, e.g., [10]. It is also worth to mention that the class of p-regular languages was one of the first subclasses of the regular languages for which the star height problem was shown to be decidable, see, e.g., [1]. Recently, the family of p-regular languages, and thus PFAs, gained renewed interest. For instance, in [9] the decomposing of PFAs into the intersection of smaller automata of the same kind was investigated. Moreover the operational state complexity on PFAs was studied in [7]. Up to our knowledge the operational accepting state complexity of p-regular languages was not investigated so far. We study this problem by examining the following question:

> Given are three non-negative integers m, n, and α and a regularity preserving language operation $\circ$, are there minimal permutation automata A and B with accepting state complexity m and n, respectively, such that the language $L(A) \circ L(B)$ is accepted by a minimal deterministic finite automaton with α accepting states?

Following the terminology of [8] we call values α "magic" if there are no such automata A and B. The following results were shown in [2] and [6] for the operational accepting state complexity on languages accepted by DFAs—for accepting state complexities m and n one can obtain all values from the given number set for α:

- Complement: $\mathbb{N} \cup \{\, 0 \mid m = 1 \,\}$.
- Kleene star and Kleene plus: $\mathbb{N}$.
- Union: $\mathbb{N}$.
- Set difference: $\mathbb{N}$.
- Intersection: $[0, mn]$ (if the input alphabet is at least binary).
- Reversal: $\mathbb{N}$.
- Quotient: $\mathbb{N} \cup \{0\}$.

One may have noticed that for none of the above mentioned operations magic numbers exist up to the special case of complementation and $m = 1$.[1] It is worth mentioning that these results were obtained without the use of PFAs automata witnesses.

We generalize these results for the operations complement, union, set difference, Kleene star, and Kleene plus to the family of languages accepted by PFAs. This means even though PFAs are restricted in their expressive power compared to arbitrary DFAs, there are no magic numbers for the accepting state complexity of PFAs w.r.t. the above mentioned operations. When considering the reversal operation a significant difference appears. While for DFAs the reversal operation induces the whole set $\mathbb{N}$ as accepting state complexities as mentioned above, this is not the case for PFAs, where we can prove that the number $\alpha = 1$ is magic for every $m \geq 2$. In fact, we prove that for $m = 2$ no other magic number as $\alpha = 1$ exists. Whether this is also true for larger m is left open. Yet another difference in the accepting state complexity comes from the quotient operation. Here it turns out that for unary languages accepted by PFAs only the range $[1, mn]$ is obtainable for the accepting state complexity. This is entirely different compared to the general case. Finally, the unary case for the accepting state complexity of the intersection operation for DFAs in general was left open in [6]. We close this gap by considering this

[1]The intersection of two languages L_1 and L_2 of accepting state complexity m and n is accepted by the cross product of the minimal DFAs accepting L_1 and L_2. So the accepting state complexity is directly bounded by mn. Therefore the numbers greater mn are not of interest.

problem in detail. In this way, we identify a whole range of magic numbers for the intersection of unary languages accepted by PFAs and extend this result to the case of DFAs, solving the left open problem mentioned above. Due to space constraints some of the proofs are omitted; they can be found in the full version of this paper.

2 Preliminaries

Let $\mathbb{N}$ denote the set of all positive integers and $\mathbb{N}_{\geq x}$ ($\mathbb{N}_{\leq x}$, respectively) the set of all positive integers that are greater or equal x (less or equal x, respectively).

We recall some definitions on finite automata as contained in [4]. Let Σ^* denote the set of all words over the finite alphabet Σ. The *empty word* is denoted by ε. Further, we denote the set $\{i, i+1, \ldots, j\}$ by $[i, j]$, if i and j are integers. A *deterministic finite automaton* (DFA) is a quintuple $A = (Q, \Sigma, \cdot, s, F)$, where Q is the finite set of *states*, Σ is the finite set of *input symbols*, $s \in Q$ is the *initial state*, $F \subseteq Q$ is the set of *accepting states*, and the *transition function* $\cdot$ maps $Q \times \Sigma$ to Q. The *language accepted* by the DFA A is defined as

$$L(A) = \{\, w \in \Sigma^* \mid s \cdot w \in F \,\},$$

where the transition function is recursively extended to a map $Q \times \Sigma^*$ onto Q. Obviously, every letter $a \in \Sigma$ induces a mapping on the state set Q to Q by $q \mapsto q \cdot a$, for every $q \in Q$. A DFA is *unary*, if the input alphabet Σ is a singleton set, that is, $\Sigma = \{a\}$, for some input symbol a. Moreover, if every letter of the automaton induces only permutations on the state set, then we simply speak of a *permutation* automaton (PFA).

As usual we denote the *state complexity* of a language L accepted by a DFA by

$$\mathrm{sc}(L) = \min\{\, \mathrm{sc}(A) \mid A \text{ is a DFA with } L = L(A) \,\},$$

where $\mathrm{sc}(A)$ refers to the number of states of the automaton A. Similarly we define the measure $\mathrm{asc}(L)$ the *accepting state complexity* of a language L accepted by a DFA, where $\mathrm{asc}(A)$ refers to the number of final states of the automaton A.

An automaton is *minimal* (a-minimal, respectively) if it admits no smaller equivalent automaton w.r.t. the number of states (final 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 [2] applies:

Theorem 1. *Let L be a language accepted by a minimal DFA A. Then* $\mathrm{asc}(L) = \mathrm{asc}(A)$.

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

$$g^c_\circ(n_1, n_2, \ldots, n_k)$$

as the set of all integers r 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 $c(\circ(L_1, L_2, \ldots, L_k)) = r$. In case we only consider unary languages $L_1, L_2, \ldots, L_k$ we simply write $g^{c,u}_\circ$ instead. When restricting the underlying languages to, e.g., be accepted by permutation automata (PFAs), we indicate this by writing $g^c_{\circ,\mathrm{PFA}}$ and $g^{c,u}_{\circ,\mathrm{PFA}}$, respectively.

In order to explain the notation we give a small example.

Example 2. *Consider the unary operation C of complementation of languages. It is obvious that*

$$g^{\mathrm{sc}}_C(m) = \{m\}, \quad \textit{for } m \geq 1.$$

On the other hand, when we consider the accepting state complexity, in [2] the following behaviour

$$g_C^{\mathrm{asc}}(m) = \begin{cases} \{1\} & \text{if } m = 0, \\ \{0\} \cup \mathbb{N} & \text{if } m = 1, \\ \mathbb{N} & \text{otherwise,} \end{cases}$$

for the complementation was proven. Moreover, it is easy to see that

$$g_C^{\mathrm{sc},u}(m) = g_C^{\mathrm{sc}}(m) \quad \text{and} \quad g_C^{\mathrm{asc},u}(m) = g_C^{\mathrm{asc}}(m)$$

holds. □

In the constructions to come, note that we will use the mod operation such that $x \bmod y+z$ is the same as $(x \bmod y)+z$ and not equal to $x \bmod (y+z)$, but $x+y \bmod z$ is the same as $(x+y) \bmod z$. We use $\div$ for the integer division and $/$ for the ordinary division.

3 Results

We investigate the accepting state complexity of various regularity preserving language operations such as union, quotient, complement, difference, intersection, Kleene star, Kleene plus, and reversal on languages accepted by permutation automata. Before we start our investigation we introduce a useful notion for *unary* permutation automata by strings. Since a unary permutation automaton consists of a cycle only, it suffices to encode the finality of these cycle states by a binary string. This is done as follows: a word $w \in \{0,1\}^+$ with $w = a_0 a_1 \ldots a_{|w|-1}$, for $a_i \in \{0,1\}$ and $0 \leq i \leq |w|-1$, describes the permutation automaton

$$A_w = (\{0,1,\ldots,|w|-1\}, \{a\}, \cdot, 0, \{\, i \mid 0 \leq i < |w|-1 \text{ and } a_i = 1 \,\})$$

with

$$i \cdot a = \begin{cases} i+1 & \text{for } 0 \leq i < |w|-1, \\ 0 & \text{otherwise.} \end{cases}$$

It is clear that there is a bijection between words in $\{0,1\}^+$ with all unary PFAs. Thus, we can identify words with PFAs and *vice versa*. Now we are ready for the investigation of the accepting state complexity of certain operations on PFAs.

3.1 Complementation

The complement of a language accepted by a finite automaton can be obtained by simply interchanging accepting and non-accepting states. Hence, the state complexity of a language accepted by a finite automaton, regardless whether the automaton is a permutation automaton or not, is the same. The result on the *accepting state complexity* for unrestricted DFAs was presented in Example 2. Next we show that this result even holds for PFAs.

Theorem 3. *We have* $g_{C,PFA}^{\mathrm{asc},u}(m) = g_{C,PFA}^{\mathrm{asc}}(m) = g_C^{\mathrm{asc}}(m) = g_C^{\mathrm{asc},u}(m)$.

3.2 Kleene Star and Kleene Plus

Next we study the accepting state complexity of the Kleene star and the Kleene plus operations for permutation automata. We want to mention that the Kleene closure of a p-regular language cannot be accepted by a PFA in many cases, see for example [7]. First we prove a useful relation between PFAs and the languages which they accept.

Lemma 4. *Let I be a finite set of non-negative integers and j be a number which is greater than the biggest number in I. The language $\bigcup_{i\in I} a^i(a^j)^*$ can be accepted by the PFA*

$$A = (\{q_0, q_1, \ldots, q_{j-1}\}, \{a\}, \cdot_A, q_0, \{q_i \mid i \in I\})$$

with $q_i \cdot_A a = q_{i+1 \bmod j}$. Additionally A is minimal if there is no divisor t of j such that for every $i \in I$ the number $i+t \bmod j$ is in I.

Proof. The tedious details for the first statement are left to the reader. We prove the second statement by contradiction. So assume that there is a divisor t of j such that for every $i \in I$ the number $i+t \bmod j$ is in I. Assuming that A is minimal, for every pair of states there is a word w which distinguishes them, i.e., maps one of the states onto an accepting and the other one onto a non-accepting state. This includes the states q_i and $q_{i+t \bmod j}$. Since $w \in a^*$ we can assume $w = a^k$ for a non-negative integer k. Due to the definition of A the word w maps the states q_i and $q_{i+t \bmod j}$ onto $q_{i+k \bmod j}$ and $q_{i+t+k \bmod j}$ which are either both in I or are both not in I. This contradicts the assumption that w distinguishes q_i and $q_{i+t \bmod j}$. ☐

We will use this result to prove that no magic numbers exist for the accepting state complexity of the Kleene star operation.

Theorem 5. *We have*

$$g_*^{\mathrm{asc},u}(m) = g_{*,PFA}^{\mathrm{asc},u}(m) = \begin{cases} \{1\} & \text{if } m = 0, \\ \mathbb{N} & \text{otherwise.} \end{cases}$$

Proof. For $m = 0$ we observe that $\emptyset^* = \{\varepsilon\}$, for ε being the empty word. So the first statement follows. For the second one we distinguish whether α or m are equal to one. We distinguish four cases, where in each case we use Lemma 4 to show that the defined language has accepting state complexity m:

1. Case $\alpha = 1$ and $m > 1$: The language $L = \bigcup_{i=1}^{m} a^i(a^{m+1})^*$ has accepting state complexity m and its Kleene star is equal to Σ^* which has accepting state complexity one.
2. Case $\alpha = 1$ and $m = 1$: The language of the previous case can also be used if $m = 1$.
3. Case $\alpha > 1$ and $m > 1$: Let $L = a^2(a^{2(\alpha-1)+m+1})^* \cup \bigcup_{i=1}^{m-1} a^{2(\alpha-1)+i}(a^{2(\alpha-1)+m+1})^*$. Then L^* is equal to $\bigcup_{i=0}^{\alpha-2} a^{2i} \cup a^{2(\alpha-1)}\Sigma^*$. In turn $\bigcup_{i=0}^{\alpha-2} a^{2i} \cup a^{2(\alpha-1)}\Sigma^*$ can be accepted by a unary DFA which has a tail of length $2(\alpha-1)$ and a cycle formed by one state, where all states on positions with an even number are accepting if we start counting by zero.
4. Case $\alpha > 1$ and $m = 1$: Define $L = a^2(a^{2\alpha-1})^*$. Then L^* is equal to $\bigcup_{i=0}^{\alpha-2} a^{2i} \cup a^{2(\alpha-1)}\Sigma^*$, which is the Kleene star language from the previous case.

So in all cases the Kleene closure has accepting state complexity α which completes the proof. ☐

By taking into account that for every language L the empty word ε is in L^+ if and only if $\varepsilon \in L$, with a small adjustment of the used languages for the previous theorem we obtain the following corollary for the Kleene plus operation.

Corollary 6. *We have*

$$g_{+}^{\mathrm{asc},u}(m) = g_{+,PFA}^{\mathrm{asc},u}(m) = \begin{cases} \{0\} & \text{if } m = 0, \\ \mathbb{N} & \text{otherwise.} \end{cases}$$

3.3 Union

In this subsection we extent the results for the accepting state complexity from [2] for the union operation to the class of permutation automata. For DFAs in general the following result was shown in [2]:

$$g_{\cup}^{\mathrm{asc},u}(m,n) = g_{\cup}^{\mathrm{asc}}(m,n) = \begin{cases} \{m\} & \text{if } n = 0, \\ \{n\} & \text{if } m = 0, \\ \mathbb{N} & \text{otherwise,} \end{cases}$$

and since the union operation is commutative $g_{\cup}^{\mathrm{asc},u}(m,n) = g_{\cup}^{\mathrm{asc},u}(n,m)$ and $g_{\cup}^{\mathrm{asc}}(m,n) = g_{\cup}^{\mathrm{asc}}(n,m)$. Note the languages that prove these results are *not* accepted by any PFA.

We will prove that except for the special cases $m = 0$ or $n = 0$ every accepting state complexity can be reached also for unary alphabets. Therefore the reachable numbers coincide in the cases that the input DFAs are restricted or not. We split this into three theorems which show that $\mathbb{N}_{\leq \min\{n,m\}}$, $\mathbb{N}_{\geq \max\{n,m\}}$ and $[\min\{n,m\}+1, \max\{n,m\}-1]$ are reachable. We start with the upper range.

Theorem 7. *For $m,n \geq 1$ we have $\mathbb{N}_{\geq \max\{n,m\}} \subset g_{\cup,PFA}^{asc,u}(m,n)$.*

We split the proof of this theorem into two lemmata, which show the reachability of smaller intervals (Lemma 8) and that the union of those intervals equals the whole range $\mathbb{N}_{\geq \max\{n,m\}}$ (Lemma 9).

Lemma 8. *Let $m \geq n \geq 1$, $i \geq 1$ and $\alpha \in [\max\{in,m\}, in+m]$. There are minimal unary PFAs A and B with accepting state complexity m and n, respectively, such that $L(A) \cup L(B)$ has accepting state complexity α.*

The next lemma shows that the union of the intervals which are reachable due to the previous lemma is again an interval.

Lemma 9. *For $m \geq n$ holds $\bigcup_{i\in\mathbb{N}}[\max\{in,m\}, in+m] = \mathbb{N}_{\geq m}$.*

It is not hard to see that the Lemmata 8 and 9 are symmetric in n and m so together they prove Theorem 7. Next we show that the lower range is reachable, too.

Theorem 10. *For $m,n \geq 1$ we have $[1, \min\{n,m\}] \subset g_{\cup,PFA}^{asc,u}(m,n)$.*

The constructions for the previous lemmata created sequences $1^\alpha 0^\ell$ for some $\ell \in \mathbb{N}$. There are values for n, m and α such that α cannot be reached by this method. Therefore we have to create sequences which contain α accepting and distinguishable states which are not consecutive. We want to mention here that we count the positions of the states in an unary PFA in the same way we count the positions of the letters describing the PFA, namely we start by zero.

Theorem 11. *For $m,n \geq 1$ we have $[\min\{n,m\}+1, \max\{n,m\}-1] \subset g_{\cup,PFA}^{asc,u}(m,n)$.*

Obviously $K \cup \emptyset = K$ and $\emptyset \cup L = L$ for every languages $K, L \subseteq \Sigma^*$. Together with the Theorems 7, 10, and 11 we obtain the following corollary.

Corollary 12. *We have*

$$g_{\cup}^{asc,u}(m,n) = g_{\cup,PFA}^{asc,u}(m,n) = \begin{cases} \{n\} & \text{if } n = 0, \\ \{m\} & \text{if } m = 0, \\ \mathbb{N} & \text{otherwise.} \end{cases}$$

3.4 Difference

Now let us come to the difference operation which was also considered in [2]. For deterministic finite automata with no restrictions the following result was shown:

$$g_{\setminus}^{\mathrm{asc},u}(m,n) = g_{\setminus}^{\mathrm{asc}}(m,n) = \begin{cases} \{m\} & \text{if } n = 0 \\ \{0\} & \text{if } m = 0 \\ \{0\} \cup \mathbb{N} & \text{otherwise,} \end{cases}$$

Again the languages that prove these results are *not* accepted by any PFA.

This subsection is structured as follows. First we will use the fact that $K \setminus L = K \cap C(L)$ for all finite languages K and L to show that all numbers in the range $[0,m]$ are reachable and for all $n,m \geq 1$ with $\alpha \bmod m = 0$ the numbers α are reachable as well. The previously mentioned fact for the set difference of two languages allows us to prove the first two statements by constructing A and B such that the minimal DFA accepting the language of the direct product of A and $\overline{B}$ has the required size, for $\overline{B}$ being equal to B except that its set of accepting states is complemented. Afterwards we prove that for all $n,m \geq 1$ with $\alpha \bmod m > 0$ the numbers α are reachable, too.

Lemma 13. *For $m,n \geq 1$ we have $[0,m] \subset g_{\setminus,PFA}^{asc,u}(m,n)$.*

Proof. Let $A = A_w$ and $B = A_{w'}$ for $w = (0^n1)^{\alpha}(10^n)^{m-\alpha}0^{n+1}$ and $w' = 1^n0^1$. That means $L(A) \setminus L(B)$ is accepted by $A_{w''}$ for $w'' = (0^n1)^{\alpha}0^{(n+1)\cdot(m-\alpha+1)}$. We leave it to the reader to observe that the three involved automata are minimal. The PFA $A_{w''}$ has accepting state complexity α which proves the statement of the lemma. □

The next lemma shows that the every number α in the upper range $\mathbb{N}_{\geq m+1}$ is obtainable if α fulfills $\alpha \bmod m = 0$.

Lemma 14. *For $m,n \geq 1$ and $\alpha \in \mathbb{N}_{\geq m+1}$ with $\alpha \bmod m = 0$ we have $\alpha \in g_{\setminus,PFA}^{asc,u}(m,n)$.*

Proof. Let $\alpha = mx + (\alpha \bmod m)$ and we set $A = A_w$ and $B = A_{w'}$ for $w = 1^m0^k$ and $w' = 0^x1^n$, where k is the smallest positive integer such that $\gcd(m+k, x+n) = 1$. It is not hard to see that A and B are minimal and that the cross product DFA of A and $\overline{B} = A_{w''}$, for $w'' = 1^x0^n$, has xm accepting states and is minimal, too. □

The next lemma shows that the upper interval $\mathbb{N}_{\geq m+1}$ is also attainable for $\alpha \bmod m \neq 0$ which clearly proves the range to be reachable for all numbers in the interval.

Lemma 15. *For $m,n \geq 1$ and $\alpha \in \mathbb{N}_{\geq m+1}$ with $\alpha \bmod m \neq 0$ we have $\alpha \in g_{\setminus,PFA}^{asc,u}(m,n)$.*

By combining the Lemmata 13, 14, and 15 we obtain the following corollary.

Corollary 16. *We have*

$$g^{\mathrm{asc},u}_{\setminus}(m,n) = g^{asc,u}_{\setminus,PFA}(m,n) = \begin{cases} \{m\} & \text{if } n=0,\\ \{0\} & \text{if } m=0,\\ \{0\}\cup\mathbb{N} & \text{otherwise.}\end{cases}$$

Proof. Since $K\setminus\emptyset = K$ and $\emptyset\setminus L=\emptyset$ for every languages $K,L\subseteq\Sigma^*$ the first two statements follow immediately. Additionally the last statement follows by the Lemmata 13, 14, and 15. □

3.5 Intersection

We show that the left open unary case for the intersection operation for both PFAs and DFAs differs from the solved general case. It is not hard to see that at most the numbers in the range $[1,nm]$ can be reached. We split this interval into three smaller ones, namely $[0,\max\{n,m\}]$, $[\max\{n,m\}+1, nm-\min\{n,m\}]$ and $[nm-\min\{n,m\}+1,nm]$.

Lemma 17. *We have* $[0,\max\{n,m\}]\subseteq g^{\mathrm{asc},u}_{\cap,PFA}(m,n)$.

Proof. Since for the intersection operation the ordering of the input languages is irrelevant we assume $m\geq n$. Let $A=A_w$ and $B=A_{w'}$ for $w=(10^n)^\alpha(0^n1)^{m-\alpha}0^{n+1}$ and $w'=1^n0$, respectively. Both PFAs are minimal since the last $n+1$ states of A do not contain an accepting state and the accepting states of B form a sequence. The minimal DFA accepting the language $L(A)\cap L(B)$ is the PFA $A_{w''}$ for $w''=(10^n)^\alpha 0^{(n+1)\cdot(m-\alpha+1)}$ which obviously has accepting state complexity α. □

We found out by exhaustive search that for small m and n the following conjecture holds.

Conjecture 18. *All numbers in* $[\max\{n,m\}+1, nm-\min\{n,m\}]$ *which are* not *in*

$$\begin{aligned}&[\max\{n,m\},n+m]\cup\\ &\qquad\{\,t_nx_m \mid t_n \text{ is a nonzero divisor of } n \text{ and } 0\leq x_m\leq (nm-\min\{n,m\})\div t_n\,\}\cup\\ &\qquad\qquad\{\,t_mx_n \mid t_m \text{ is a nonzero divisor of } m \text{ and } 0\leq x_n\leq (nm-\min\{n,m\})\div t_m\,\}\end{aligned}$$

are magic.

Next we investigate the numbers in the range $[nm-\min\{n,m\}+1,nm]$. For showing that all numbers except nm are magic we prove the following structural property of the cross product for PFAs.

Lemma 19. *Let q_0,q_1 and p_0,p_1 be states of the minimal unary PFAs A and B, respectively. If (q_0,p_0), (q_1,p_0) and (q_0,p_1) are initially reachable in the cross product automaton C then (q_1,p_1) is initially reachable, too.*

Proof. Let $\Sigma=\{a\}$ be the input alphabet of A, B, and C. Since (q_0,p_0), (q_1,p_0), and (q_0,p_1) are initially reachable in C we know that there are words w_q and w_p which map (q_0,p_0), onto (q_1,p_0) and (q_0,p_0), onto (q_0,p_1). Because A and B are unary we observe that w_q and w_p induce the identity on B and A, respectively. This implies that $(q_0,p_0)\cdot w_qw_p=(q_1,p_0)\cdot w_p=(q_1,p_1)$ which proves the stated claim. □

One may ask whether Lemma 19 holds for alphabets of arbitrary size. In general it is not true that w_q and w_p induce the identity on B and A, respectively. Instead those words induce a cycle on B and A that has a size that divides the order of the word. If the cycle of A and B contain q_0, q_1 and p_0, p_1, respectively, the statement of the Lemma remains true. We leave it to the reader to prove or disprove the lemma above for alphabets of at least two letters. As mentioned before we use Lemma 19 to prove that a whole range of numbers in the upper interval cannot be reached.

Theorem 20. *We have* $[nm - \min\{n,m\}+1, nm-1] \not\subseteq g^{\mathrm{asc},u}_{\cap,PFA}(m,n)$.

Proof. Let $\alpha \in [nm - \min\{n,m\}+1, nm-1]$. Clearly Lemma 19 implies that for all PFAs A and B their cross product automaton has either less than $nm - \min\{n,m\}+1$ or nm initially reachable accepting states. On the other hand a result from [5, Lemma 4] implies that if a PFA has nm accepting states and it is not minimal then its minimal DFA has t accepting states for a divisor of nm. Since every divisor of nm is less than $nm - \min\{n,m\}+1$ the claim of the theorem follows. □

If we look at the cross product automaton C of unary DFAs A and B we see that for every state q that is in the tail of A or B there is exactly one initially reachable state in C which contains q as one of its components. So we obtain that C contains at most $nm - \min\{n,m\}+1$ initially reachable accepting states. Together with Theorem 20 we obtain the following corollary.

Corollary 21. *We have* $[nm - \min\{n,m\}+2, nm-1] \not\subseteq g^{\mathrm{asc},u}_{\cap}(m,n)$.

As mentioned before the upper bound for $g^{\mathrm{asc},u}_{\cap,\mathrm{PFA}}(m,n)$ is not a magic number which is proven in the following lemma.

Lemma 22. *We have* $nm \in g^{\mathrm{asc},u}_{\cap,PFA}(m,n)$.

Proof. Let $A = A_w$ and $B = A_{w'}$ for $w = 1^m0^n$ and 1^n0^{m+1}, respectively. Since $n+m$ and $n+m+1$ are coprime it is obvious that length of their product automaton C is $(n+m)\cdot(n+m+1)$. So each pair of accepting states is initially reachable. We observe that there are $\max\{n,m\} - \min\{n,m\}$ sequences of accepting states of length $\min\{n,m\}$. All of those sequences follow each other, i.e., only non-accepting states are between them. There are also shorter sequences of accepting states in C, e.g., an accepting states which follows and is followed by a non-accepting state. This implies that C has to be minimal which proves the stated claim. □

3.6 Reversal

The results of this subsection are in contrast to the general case where arbitrary DFAs are considered. Here the restriction for the input automaton to be a PFA provides magic numbers which are not magic if the input automaton is not restricted. For deterministic finite automata with no restrictions the following result was proven in [6]:

$$g^{\mathrm{asc}}_{R}(m,n) = \begin{cases} \{0\} & \text{if } m = 0, \\ \mathbb{N} & \text{otherwise.} \end{cases}$$

We show that in the case of permutation automata the number $\alpha = 1$ is magic for all $m \geq 2$. Before we do this we need a special PFA that plays an important role for the reversal operation. We want to mention that for a unary language L its reversal L^R is equal to L. So we will only consider languages with at least two different letters. First we define $\binom{S}{k}$ for a finite set S and a non-negative integer to be the set of all subsets of S which have size k. For a PFA $A = (\{q_0, q_1, \ldots, q_{k-1},\}, \Sigma, \cdot_A, q_0, F_A)$, we define

$$A_R = (\binom{Q_A}{|F_A|}, \Sigma, \cdot_{A_R}, F_A, \{R \in \binom{Q_A}{|F_A|} \mid q_0 \in R\}),$$

where $R \cdot_{A_R} w = \{q \cdot_A w^{-1} \mid q \in R, w \in \Sigma\}$ for all $R \in \binom{Q_A}{|F_A|}$. We want to mention here that A_R is a well-defined DFA since for every word w the mapping w^{-1} is uniquely defined because A is a PFA. Because w^{-1} applies the reverse transitions to every state of A and every state of A_R that contains the initial state of A is accepting so it is not hard to see that A_R accepts the language $L(A)^R$. Before we prove our results for the accepting state complexity of the reversal operation of PFAs we derive two structural properties of the DFA A_R. First we count the number of initially reachable states in A_R.

Lemma 23. *Let A be a minimal PFA. Then there is an integer $x \geq 1$ such that for every state q of A there are x initially reachable states in A_R containing q.*

Proof. Let q be an arbitrary state of A. Assume there are $x \geq 1$ states $R_0, R_1, \ldots, R_{x-1}$ in A_R which contain q. Since A is an PFA the images of those states are different regardless of the choice of the mapping. If we apply the mapping which maps q onto q' for any other state q' of A it follows directly that there are at least x states of A_R which contain q'. Since this argument can be used symmetrically the claim of the lemma follows. □

Since the states of A_R are in turn sets we prove the following property of A_R which is the automata theoretical interpretation of the fact that bijections on elements induce bijections on sets of those elements.

Lemma 24. *For every PFA A the DFA A_R is a PFA, too.*

Proof. Since A is a PFA for every letter a the preimage of any state q of A is uniquely defined. By applying this property to every state q in a state R of A_R we directly obtain the unique preimage of R. □

Now we will prove our magic number result for the PFA case of the reversal operation.

Lemma 25. *Let $m \geq 2$. Then there exists* no *PFA A with $asc(A) = m$ such that* $\operatorname{asc}(A_R) = 1$.

The result of the previous lemma proves the inequality statement of our main theorem for the accepting state complexity of the reversal operation of p-regular languages. Obviously we also prove that for m equal two every number unequal one is not magic. We do this by constructing an automaton A such that A_R has $\alpha \cdot k \div m = \binom{k}{m}$ initially reachable states while every state of A appears in exactly α of them.

Theorem 26. *We have*

$$g^{\mathrm{asc}}_{R,PFA}(m) = \begin{cases} \{0\} & \text{if } m = 0, \\ \{1\} & \text{if } m = 1, \\ \mathbb{N}_{\geq 2} & \text{if } m = 2, \end{cases}$$

and $g^{\mathrm{asc}}_{R,PFA}(m) \neq \mathbb{N}$ if $m \geq 3$. Therefore $g^{\mathrm{asc}}_{R,PFA}(m) \neq g^{\mathrm{asc}}_{R}(m)$.

We note here that for $m \geq 3$ the equation $\alpha \cdot k \div m = \binom{k}{m}$ has no integer solution for many values of α and m which can be easily confirmed. For those values of α and m for which the equation has an integer solution we obtain $\alpha \in g^{\mathrm{asc}}_{R,\mathrm{PFA}}(m)$ in similiar fashion like for $m = 2$. Nevertheless we conjecture the following:

Conjecture 27. *We have*

$$g^{\mathrm{asc}}_{R,PFA}(m) = \begin{cases} \{0\} & \text{if } m = 0, \\ \{1\} & \text{if } m = 1, \\ \mathbb{N}_{\geq 2} & \text{if } m \geq 2. \end{cases}$$

Clearly this would mean that $\alpha = 1$ is the only number which is magic for the reversal of p-regular languages and non-magic for arbitrary regular languages.

3.7 Quotient

For two DFAs A and B the right quotient $L(A)L(B)^{-1}$ can be accepted by the DFA $\tilde{A}$ which can be obtained from A by exchanging its set of accepting states F by $\{\, q \mid q \cdot w \in F$ for some w in $L(B) \,\}$, which we denote by $\tilde{F}$. It is obvious that $\tilde{A}$ is a PFA if A is a PFA. Additionally, if s is the initial state of A, then the automaton obtained from A by making all states in $\{\, s \cdot w \mid w \in L \,\}$ initial accepts $L(B)^{-1}L(A)$. Since for unary languages the left and right quotient coincide no distinction is made at this point and we use the right quotient unless otherwise stated. For regular languages in general the following was shown in [6]:

$$g_{-1}^{\mathrm{asc},u}(m,n) = g_{-1}^{\mathrm{asc}}(m,n) = \begin{cases} \{0\} & \text{if } m = 0 \text{ or } n = 0, \\ \{0\} \cup \mathbb{N} & \text{otherwise.} \end{cases}$$

Clearly the first statement follows directly from the fact that $K\emptyset^{-1} = \emptyset L^{-1} = \emptyset$ for all languages K and L. But we show that last statement does not hold for the class of p-regular languages. For this we distinguish whether n is equal one or at least equal to two. First we show which numbers are reachable if n equals one.

Lemma 28. *We have* $[1,m] \subseteq g_{-1,PFA}^{\mathrm{asc},u}(m,1)$.

Proof. Let α be in $[1,m]$. Define $A = A_w$ and $B = A_{w'}$ for $w = 1^{\alpha}0^{m+1-\alpha}(10^m)^{m-\alpha}0^{m+1}$ and $w' = 010^{m-1}$. We observe that A has length $(m+1)(m-\alpha+2)$ and B has length $m+1$. It is not hard to see that

$$L(A) = \{\, a^{i+x(m+1)(m-\alpha+2)} \mid 0 \le i \le \alpha - 1, 0 \le x \,\} \cup \{\, a^{(m+1)i+x(m+1)(m-\alpha+2)} \mid 1 \le i \le m-\alpha, 0 \le x \,\}$$

and

$$L(B) = \{\, a^{(m+1)i+1} \mid i \in \mathbb{N} \cup \{0\} \,\},$$

for a being the letter of the input alphabet of A and B. We observe that the PFA $\tilde{A}$ has the set of accepting states $\tilde{F}$ that contains exactly the elements $q_{(i-((m+1)j+1)) \bmod (m+1)(m-\alpha+2)}$, for $0 \le i \le \alpha - 1$ or $i \in \{\, (m+1)\ell \mid 1 \le \ell \le m - \alpha \,\}$ and $j \in \mathbb{N} \cup \{0\}$. Alternatively we can write

$$\tilde{F} = \{\, q_{(i-((m+1)j+1)) \bmod (m+1)(m-\alpha+2)} \mid 0 \le i \le \alpha - 1, j \in \mathbb{N} \cup \{0\} \,\}$$

because

$$\begin{aligned} 0 - ((m+1)(m-\alpha+2-j)+1) &\bmod (m+1)(m-\alpha+2) \\ &= (m+1)j - ((m+1)\cdot 0) + 1 \bmod (m+1)(m-\alpha+2) \end{aligned}$$

holds. One may observe that $\tilde{F}$ contains all accepting states of A but their index is decreased by one modulo the length of A. If we shift those states again by an arbitrary multiple of $(m+1)$ we obtain the remaining states in $\tilde{F}$. Clearly $\tilde{F}$ does not contain other states. Therefore $\tilde{A} = A_{w''}$, for $w'' = 1^{\alpha-1}0^{m+1-\alpha}(1^{\alpha}0^{m+1-\alpha})^{m-\alpha+1}1$. Indeed this PFA is not minimal, i.e., all of its sequences $1^{\alpha-1}0^{m+1-\alpha}1$ are equivalent. Thus the minimal PFA accepting the language $L(A)L(B)^{-1}$ is $A_{w'''}$, for the word $w''' = 1^{\alpha-1}0^{m+1-\alpha}1$ which has accepting state complexity α. □

Next we prove that every number which is not reachable due to the previous lemma is magic.

Lemma 29. *We have* $[1,m] = g_{-1,PFA}^{\mathrm{asc},u}(m,1)$.

Proof. Due to the proof of Lemma 28 it remains to show that $\mathbb{N}_{\geq m+1}$ is not in $g^{\mathrm{asc},u}_{-1,\mathrm{PFA}}(m,1)$. Therefore let A and B be unary minimal PFAs with m and n accepting states, respectively. Recall that $\tilde{A}$ is the PFA obtained from A by replacing its set of accepting states with $\tilde{F} = \{\, q \mid q \cdot w \in F \text{ for some } w \text{ in } L(B)\,\}$. We observe that the set of accepting states of $\tilde{A}$ is equal

$$\tilde{F} = \{\, q_{(i-(jk'+\ell)) \bmod k} \mid i \in I_A, j \in \mathbb{N} \cup \{0\}\,\}$$

for I_A being the index set of the accepting states of A, q_ℓ being the accepting state of B and k, k' being the number of states of A and B, respectively. For an arbitrary but fixed $i \in I_A$ we see that each of the states

$$q_{(i-(0k'+\ell)) \bmod k}, q_{(i-(1k'+\ell)) \bmod k}, q_{(i-(2k'+\ell)) \bmod k}, \ldots$$

can be mapped by $a^{k'}$ onto its predecessor. Since this holds for every $i \in I_A$ those states have to be equivalent which proves that $\tilde{A}$ contains at most m inequivalent accepting states. $\square$

Now we generalize Lemma 28, for $n \geq 2$.

Lemma 30. *We have* $[1,mn] \subseteq g^{\mathrm{asc},u}_{-1,PFA}(m,n)$*, for* $n \geq 2$.

Next we rule out every number that is not reachable by Lemma 30. Like in the proof of Lemma 29 one observes that the set of accepting states of the DFA accepting the quotient language of two p-regular languages L_1 and L_2 of accepting state complexity m and n, respectively, is given by applying the following two steps. First the accepting states of the minimal DFA accepting L_1 are shifted onto n positions. Afterwards these mn accepting states are cyclic replicated by the length of the DFA accepting L_1. Since the DFA accepting the quotient language is a PFA all cyclic replications are equivalent.

Lemma 31. *We have* $[1,mn] = g^{\mathrm{asc},u}_{-1,PFA}(m,n)$.

By using the previous four Lemmata 28, 29, 30, and 31 we deduce the following corollary.

Corollary 32. *We have*

$$g^{\mathrm{asc},u}_{-1,PFA}(m,n) = \begin{cases} \{0\} & \text{if } m = 0 \text{ or } n = 0, \\ [1,mn] & \text{otherwise.} \end{cases}$$

Therefore $g^{\mathrm{asc},u}_{-1,PFA}(m,n) \neq g^{\mathrm{asc},u}_{-1}(m,n)$.

The accepting state complexity for the quotient operation on languages accepted by permutation automata with larger input alphabets has to be left open and is subject to further research.

References

[1] J. Brzozowski (1980): *Open Problems About Regular Languages*. In R. V. Book, editor: *Formal Language Theory*, Academic Press, pp. 23–47, doi:10.1016/B978-0-12-115350-2.50007-5.

[2] J. Dassow (2016): *On the Number of Accepting States of Finite Automata*. *J. Autom., Lang. Comb.* 21(1–2), pp. 55–67, doi:10.25596/jalc-2016-055.

[3] Y. Gao, N. Moreira, R. Reis & S. Yu (2016): *A Survey on Operational State Complexity*. *J. Autom., Lang. Comb.* 21(4), pp. 251–310, doi:10.25596/jalc-2016-251.

[4] M. A. Harrison (1978): *Introduction to Formal Language Theory*. Addison-Wesley.

[5] M. Holzer & C. Rauch (2021): *The Range of State Complexities of Languages Resulting From the Cascade Product—The Unary Case*. In S. Maneth, editor: *Proceedings of the 25th International Conference on Implementation and Application of Automata*, *LNCS* 12803, Springer, Bremen, Germany, pp. 90–101, doi:10.1007/978-3-030-79121-6_8.

[6] M. Hospodár & M. Holzer (2018): *The Ranges of Accepting State Complexities of Languages Resulting From Some Operations*. In C. Campeanu, editor: *Proceedings of the 23th Conference on Implemenation and Application of Automata*, *LNCS* 10977, Springer, Charlottetown, Prince Edward Island, Canada, pp. 198–210, doi:10.1007/978-3-319-94812-6_17.

[7] M. Hospodár & P. Mlynárčik (2020): *Operations on Permutation Automata*. In N. Jonoska & D. Savchuk, editors: *Proceedings of the 24th International Conference on Developments in Language Theory*, *LNCS* 12086, Springer, Tampa, Florida, USA, pp. 122–136, doi:10.1007/978-3-030-48516-0_10.

[8] K. Iwama, Y. Kambayashi & K. Takaki (2000): *Tight bounds on the number of states of DFAs that are equivalent to n-state NFAs*. *Theoret. Comput. Sci.* 237(1–2), pp. 485–494, doi:10.1016/S0304-3975(00)00029-3.

[9] I. Jecker, N. Mazzocchi & P. Wolf (2021): *Decomposing Permutation Automata*. In S. Haddad & D. Varacca, editors: *Proceedings of the 32nd International Conference on Concurrency Theory*, *LIPIcs* 203, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, Virtual Conference, pp. 18:1–18:19, doi:10.4230/LIPIcs.CONCUR.2021.18.

[10] H. P. Zeiger (1967): *Yet Another Proof of the Cascade Decomposition Theorem for Finite Automata*. *Math. Syst. Theory* 1(3), pp. 225–228, doi:10.1007/BF01703821.