



# The Range of State Complexities of Languages Resulting from the Cascade Product—The General Case (Extended Abstract)

Markus Holzer<sup>(✉)</sup> and Christian Rauch

Institut für Informatik, Universität Giessen, Arndtstr. 2, 35392 Giessen, Germany  
{holzer,christian.rauch}@informatik.uni-giessen.de

**Abstract.** We continue our investigation on the descriptonal complexity of the cascade product of finite state devices started in [M. HOLZER, C. RAUCH: The Range of State Complexities of Languages Resulting from the Cascade Product—The Unary Case (Extended Abstract). Proc. CIAA, 2021]. Here we study the general case, that is, cascade products of reset, permutation-reset, permutation, and finite automata in general, where the left operand automaton has an alphabet of size at least two. In all cases, except for the cascade product of two permutation automata, it is shown that the whole range of state complexities, namely the interval  $[1, nm]$ , where  $n$  is the state complexity of the left operand and  $m$  that of the right one, is reachable. The cascade product of two permutation automata produces a lot of non-reachable numbers—numbers of this kind are called magic in the relevant literature—even for arbitrary alphabet sizes. These results are in sharp contrast to the unary case.

## 1 Introduction

The Krohn-Rhodes Theorem [2] states that any finite automaton can be decomposed into (several) simple “automata prime factors.” Here simple means permutation-reset automata, that is, devices where each letter induces either a permutation or a constant function on the state set. The decomposition operation is that of the cascade product, which shares similarity with the direct product of automata. Although the descriptonal complexity of the Krohn-Rhodes Theorem is well understood [11, 12], the one-time application of the cascade product operation still lacks a descriptonal complexity investigation until recently. In [6] the descriptonal complexity of the cascade product, where the left operand is a unary automaton, is studied in detail. There a complete picture on the reachable state complexities for the cascade product of reset (RFA), permutation (PFA), permutation-reset (PRFA), and finite automata in general (DFA) is drawn. It

turned out, that in the majority of the cases—in 7 out of 12 cases<sup>1</sup>—studied in [6], state complexities were identified that cannot be reached by the application of a single cascade operation on a minimal unary  $n$ - and a minimal  $m$ -state finite automaton of a certain kind. This research falls into the line of operation problems on finite state devices, see, e.g., [3, 4, 7, 8, 10], and their descriptive complexity. Adapting the notion of [9] on the determinization of nondeterministic finite automata, state numbers that cannot be reached by a binary operation on two state devices of a particular size are called “magic.” For instance, for a minimal unary  $n$ -state PFA  $A$  and a minimal  $m$ -state PFA  $B$  the minimal automaton accepting the language  $L(A \circ B)$ , where  $\circ$  refers to the cascade product, can only have  $\alpha$  states with  $\alpha$  in

$$\{1\} \cup \{nx \mid 1 \leq x \leq m\} \cup \{tx \mid 1 \leq x < m\},$$

where  $t$  is a non-trivial divisor of  $n$ , that is, a divisor that is neither 1 nor  $n$ . All other numbers in the interval  $[1, nm]$  are called magic. Interestingly, one can show that  $nm - 1$  is magic in all cases, where numbers exist that cannot be reached by the cascade operation problem where at least one automaton is not a RFA. It is worth mentioning, that the existence of these magic numbers does not contradict the Krohn-Rhodes decomposition theorem.

Here we continue the research started in [6] by considering the descriptive complexity of the cascade product of automata with input alphabets of size at least two. Compared to the unary case the situation on the existence of magic numbers completely changes for automata with larger input alphabet sizes. In almost all cases—in 15 out of 16 cases<sup>2</sup>—no magic numbers exist and thus the whole interval  $[1, nm]$  can be obtained. For the case of the cascade product of two PFAs we identify numerous magic numbers. In fact, for large  $n$  with a lot of non-trivial divisors, and small  $m$  a legion of magic numbers exist. For instance, for  $n = 10 = 2 \cdot 5$  and  $m = 3$  at least the numbers (in increasing order) 3, 7, 9, 11, 13, 17, 19, 21, 23, 27, and 29 are magic. Except for a precise characterization of the reachable state sizes for the cascade product of two PFAs in general, we solve the magic number problem for the cascade product completely.

## 2 Preliminaries

We recall some definitions on finite automata as contained in [5]. A *deterministic finite automaton* (DFA) is a quintuple  $A = (Q, \Sigma, \cdot, q_0, F)$ , where  $Q$  is the finite set of *states*,  $\Sigma$  is the finite set of *input symbols*,  $q_0 \in Q$  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  $L(A) = \{w \in \Sigma^* \mid q_0 \cdot w \in F\}$ , where the

<sup>1</sup> There are three types of automata for the left operand of the cascade product, namely unary reset, unary permutation(-reset), and unary finite automata in general and four types of automata for the right operand, that are reset, permutation, permutation-reset, and finite state device without restrictions.

<sup>2</sup> For automata with input alphabet of size at least two we have four types of left operands instead of three as in the unary case. This leads to  $4 \cdot 4 = 16$  cases.

transition function is recursively extended to a mapping  $Q \times \Sigma^* \rightarrow Q$  in the usual way. Obviously, every letter  $a \in \Sigma$  induces a mapping from the state set  $Q$  to  $Q$  by  $q \mapsto \delta(q, a)$ , for every  $q \in Q$ . A DFA is *unary* if the input alphabet  $\Sigma$  is a singleton set, that is,  $\Sigma = \{a\}$ , for some input symbol  $a$ . Moreover, a DFA is said to be a *permutation-reset* automaton (PRFA) if every input letter induces either a permutation or a constant mapping on the state set. If every letter of the automaton induces only permutations on the state set, then we simply speak of a *permutation* automaton (PFA). Finally, a DFA is said to be a *reset* automaton (RFA) if every letter induces either the identity or a constant mapping on the state set. The class of reset, permutation, permutation-reset, and deterministic automata in general are referred to as **RFA**, **PFA**, **PRFA**, and **FA**, respectively. It is obvious that inclusions  $XFA \subseteq PRFA \subseteq FA$ , where  $X \in \{\mathbf{P}, \mathbf{R}\}$ , holds. Moreover, it is not hard to see that the classes **RFA** and **PFA** are incomparable.

The cascade product [2] is originally introduced for semi-automata, which are automata with no initial nor final states. For our needs we enrich the cascade product with initial and final states and follow for the definition of the final states the lines of [1]. The *cascade product* of two DFAs  $A = (Q_A, \Sigma, \cdot_A, q_{0,A}, F_A)$  and  $B = (Q_B, Q_A \times \Sigma, \cdot_B, q_{0,B}, F_B)$ , denoted by  $A \circ B$ , is defined as the automaton

$$A \circ B = (Q_A \times Q_B, \Sigma, \cdot, (q_{0,A}, q_{0,B}), F_A \times F_B),$$

where the transition function is given by

$$(q, p) \cdot a = (q \cdot_A a, p \cdot_B (q, a)),$$

for  $q \in Q_A, p \in Q_B$ , and  $a \in \Sigma$ . We say that  $A$  is the *first automaton* and  $B$  the *second automaton* in the cascade product  $A \circ B$ . A schematic drawing of the cascade product is given in Fig. 2. It is obvious that the cascade product of two DFAs generalizes the direct product. In order to explain the notation we give an example.

*Example 1.* Consider the PRFA  $A = (\{q_0, q_1, q_2\}, \{a, b\}, \cdot_A, q_0, \{q_0, q_2\})$ , where

$$\begin{array}{lll} q_0 \cdot_A a = q_1, & q_1 \cdot_A a = q_0, & q_2 \cdot_A a = q_2, \\ q_0 \cdot_A b = q_2, & q_1 \cdot_A b = q_2, & q_2 \cdot_A b = q_2. \end{array}$$

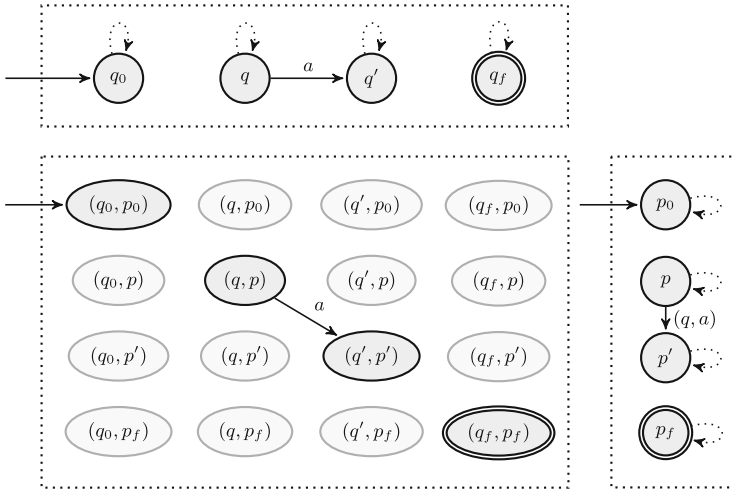
Then assume that  $m$  is an arbitrary integer greater than or equal three and let

$$B = (\{p_0, p_1, \dots, p_{m-1}\}, \{q_0, q_1, q_2\} \times \{a, b\}, \cdot_B, p_0, \{p_1\}),$$

be the PFA, where

$$\begin{array}{ll} p_i \cdot_B (q_j, b) = p_{i+1 \bmod m}, & \text{for } 0 \leq j \leq 1 \text{ and } 0 \leq i \leq m-1, \\ p_i \cdot_B (q_2, a) = p_{i+1 \bmod 3}, & \text{for } 0 \leq i \leq 2, \end{array}$$

and all other not explicitly stated transitions are self-loops. The automata  $A$  and  $B$ , for  $m = 3$ , are depicted in Fig. 1 on the top and lower right, respectively. It is easy to see that both automata are minimal.



**Fig. 1.** Schematic drawing of the cascade product of the DFAs A and B. The automaton A is depicted on top and the automaton B on the right. The automaton AB is shown in the middle, where only the transition of the state  $(q; p)$  is displayed. Note that self-loops will be only drawn by dotted loops without letters.

By construction the cascade product of  $A$  and  $B$  is given by

$$A \circ B = (\{q_0, q_1, q_2\} \times \{p_0, p_1, \dots, p_{m-1}\}, \{a, b\}, \cdot, (q_0, p_0), \{q_0, q_2\} \times \{p_1\}),$$

where the transitions of the initially reachable states

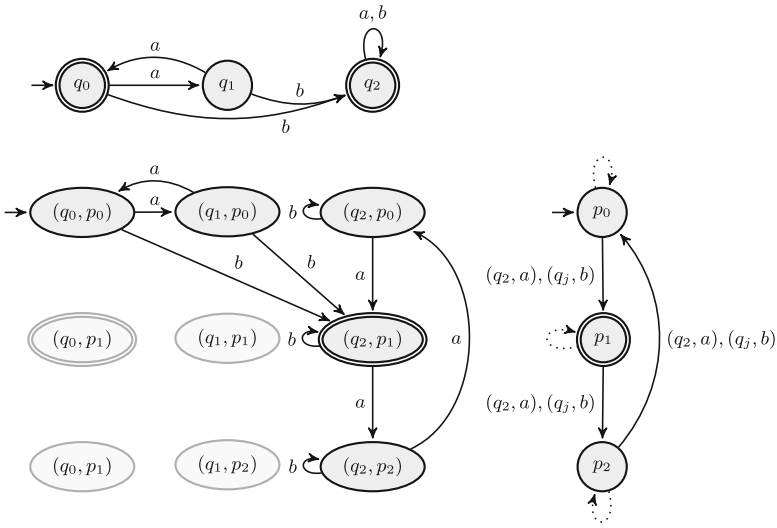
$$(q_0, p_0), (q_1, p_0), (q_2, p_0), (q_2, p_1), (q_2, p_2),$$

can be deduced from Fig. 1, too, on the lower left. Although the drawing is only for automaton  $B$  with three states, the initially reachable part of  $A \circ B$  remains the same for larger  $B$ 's as defined. Observe, that  $A \circ B$  is not a PRFA and it is not minimal. By inspection the only equivalent states in  $A \circ B$  are  $(q_0, p_0)$  and  $(q_1, p_0)$ . Hence, the minimal DFA accepting  $L(A \circ B)$  has  $\alpha = 4$  states.  $\square$

The following result is immediate by the lower bound results on the operational complexity of the intersection operation on finite automata [8].

**Theorem 1.** *Let  $A$  be a  $n$ -state and  $B$  a  $m$ -state DFA. Then  $n \cdot m$  states are sufficient and necessary in the worst case for any DFA accepting  $L(A \circ B)$ . The lower bound even holds for automata with binary input alphabet.*

When considering the descriptive complexity of the cascade product, we limit ourselves to the case where the involved automata are non-trivial, i.e., they have more than one state. This is due to the fact that if the second automaton in the operation under consideration is a singleton device, then the cascade product accepts either the empty set or the same language as the involved other device.



**Fig. 2.** The example automata A and B on the top and lower right, respectively. For a better representability not all transitions of an automaton are shown. In particular, this is the case for automaton AB, where only the transitions of the initially reachable states are shown. The cascade product AB is depicted on the lower left. Additionally the index  $j$  is a placeholder for numbers 0 and 1. Note that as before self-loops will be only depicted by dotted loops without letters.

If the first automaton is a singleton device, then the cascade product accepts either the empty language or the language  $L$  that is the image of the language that the second automaton accepts under the homomorphism  $(q, a) \mapsto a$ , for the letters  $a$  of the first automaton, where  $q$  is the state of the first automaton. Therefore, only 1-,  $n$ -, or  $m$ -state automata, for  $n, m \geq 1$ , appear as results of a cascade product with a trivial automaton. Thus, in the following we only consider non-trivial automata.

### 3 Results

We assume the reader to be familiar with the results on the cascade product of unary automata, as contained in the forerunner paper [6]. In the following we prove results for the general case, that is, if the input alphabet of the left operand automaton in the cascade product is at least two. Since we are dealing with permutation automata very often, we find the following result on the minimality of PFAs quite useful. The following statement was already shown in [6].

**Lemma 1.** *Let A be a PFA with a sole accepting state with all states reachable from the initial state. Then A is minimal. Minimality is preserved even if the initial state is changed to any other state.*

Our investigation is started with cascade products, where reset automata are involved. Observe, that all our results on the cascade product of binary automata remain valid for arbitrary alphabets of at least two letters, since adding duplicitous letters does not change the complexity.

### 3.1 Cascade Products Where Reset Automata are Involved

The magic number problem in the binary input alphabet case of the cascade product, where at least one reset automaton is involved, is already almost completely solved in [6]. It is easy to see that (minimal) reset automata form a very limited class of automata, because every minimal reset automaton has at most two states. For the cascade product of two RFAs  $A$  and  $B$  it was shown in [6] that if  $A$  has an input alphabet of at least two letters, then the minimal DFA accepting the language  $L(A \circ B)$  has  $\alpha$  states with  $1 \leq \alpha \leq 4$ , which is the maximal range for this case. This is different from the unary case, where only the values from the set  $\{1, 2, 3\}$  can be reached.

Next consider the cascade product of RFAs with PFAs, where we show that no magic numbers exist. The upper bound on the size of the minimal automaton equivalent to the cascade product of a non-trivial minimal reset automaton with a minimal  $m$ -state permutation automaton is  $2m$ . In the unary case this bound cannot be reached, since only the interval  $[1, m + 1]$  is obtained [6].

**Theorem 2.** *Let  $m \geq 1$ . Then for every  $\alpha$  with  $1 \leq \alpha \leq 2m$ , there exists a non-trivial minimal RFA  $A$  and minimal  $m$ -state PFA  $B$  such that the minimal DFA for the language  $L(A \circ B)$  has  $\alpha$  states.*

*Proof (Sketch).* By the above mentioned result in the unary case it remains to show that the integers within the interval  $[m + 2, 2m]$  are reachable. So let  $\alpha = m + \ell$ , for  $2 \leq \ell \leq m$ .

Define the RFA  $A = (\{q_0, q_1\}, \{a, b\}, \cdot_A, q_0, \{q_1\})$ , where the transition function is defined as

$$\begin{aligned} q_0 \cdot_A a &= q_1, & q_1 \cdot_A a &= q_1 \\ q_0 \cdot_A b &= q_0, & q_1 \cdot_A b &= q_1. \end{aligned}$$

It is easy to see that this automaton is minimal. Next let the PFA  $B$  be

$$B = (\{p_0, p_1, \dots, p_{m-1}\}, \{q_0, q_1\} \times \{a, b\}, \cdot_B, p_0, \{p_0\}),$$

where

$$\begin{aligned} p_i \cdot_B (q_0, a) &= p_{i+1 \bmod m}, & \text{for } 0 \leq i \leq m - 1, \\ p_i \cdot_B (q_0, b) &= p_{i+1 \bmod \ell}, & \text{for } 0 \leq i \leq \ell - 1, \\ p_i \cdot_B (q_0, b) &= p_i, & \text{for } \ell \leq i \leq m - 1, \end{aligned}$$

and

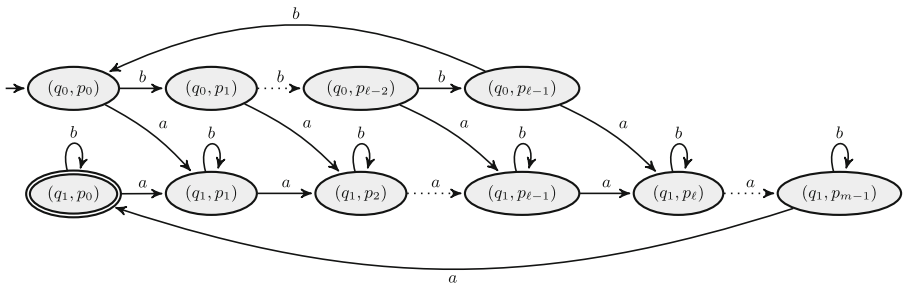
$$\begin{aligned}
 p_i \cdot_B (q_1, a) &= p_{i+1 \bmod m}, & \text{for } 0 \leq i \leq m - 1, \\
 p_i \cdot_B (q_1, b) &= p_i, & \text{for } 0 \leq i \leq m - 1.
 \end{aligned}$$

This completes the descriptions of the automaton  $B$ . By Lemma 1 the automaton  $B$  is minimal.

Next we show that the set of initially reachable states of the cascade product  $A \circ B$  can be partitioned into the union  $E_a \cup E_b$ , where

$$E_a = \{ (q_0, p_0) \cdot a^i \mid i \geq 1 \} \quad \text{and} \quad E_b = \{ (q_0, p_0) \cdot b^i \mid i \geq 0 \}.$$

Here  $\cdot$  refers to the transition function of  $A \circ B$ ; see Fig. 3. A close inspection reveals that  $E_a = \{q_1\} \times \{p_0, p_1, \dots, p_{m-1}\}$  and  $E_b = \{q_0\} \times \{p_0, p_1, \dots, p_{\ell-1}\}$ . Finally, it remains to prove that the states in  $E_a \cup E_b$  are pairwise inequivalent. The tedious details are left to the interested reader. Then the stated claim follows, because the number of states in  $E_a \cup E_b$  is  $\ell + m$ , which by construction is equal to  $\alpha$ . □



**Fig. 3.** The initially reachable part of the automaton  $A \circ B$ .

In the remainder of this section we discuss the cases where the right operand of the cascade product is an RFA. For the cascade product of a unary minimal  $n$ -state PFA with an RFA it was shown in [6] that the maximal possible interval  $[1, 2n]$  can be reached, and therefore no magic numbers exist. This result generalizes to unary PRFAs and DFAs and moreover to the non-unary case. This is summarized in the following theorem.

**Theorem 3.** *Let  $n \geq 1$ . Then for every  $\alpha$  with  $1 \leq \alpha \leq 2n$ , there exists a  $n$ -state PFA (PRFA, or DFA, respectively)  $A$  and a non-trivial RFA  $B$  such that the minimal DFA for the language  $L(A \circ B)$  has  $\alpha$  states.*

This settles all cases where RFAs are involved in the cascade product of two automata. In summary, in all cases no magic numbers exist, whenever the input alphabet of the first automaton contains at least two alphabet symbols.

### 3.2 Cascade Products of Two Permutation Automata

In [6] it was shown for the cascade product of two permutation automata in the unary case that all numbers which are relatively prime to the number of states of the first device are magic numbers. We show that this also holds in the general setting, i.e., for input alphabets of arbitrary size. Before we prove this, we recall a structural result on the cascade product of two PFAs from [6].

**Theorem 4.** *Let  $A$  and  $B$  be minimal  $n$ - and  $m$ -state PFAs, respectively. Then there is an  $x$  with  $1 \leq x \leq m$  such that for every state  $q$  in  $A$  the number of initially reachable states in  $A \circ B$  that have  $q$  as their first component is exactly  $x$ . As a direct consequence the initially reachable part of  $A \circ B$  has exactly  $nx$  states. Moreover, the minimal deterministic finite automata that accepts  $L(A \circ B)$  has  $\alpha$  states, where  $\alpha$  is a divisor of the quantity of initially reachable states of  $A \circ B$ .*

Now we are ready to prove the above mentioned result on the existence of magic numbers in the general case.

**Lemma 2.** *Let  $n, m \geq 2$ . For every  $\alpha$  in  $[2, nm]$  that is coprime to  $n$ , there does not exist a minimal  $n$ -state PFA  $A$  and a minimal  $m$ -state PFA  $B$  such that the minimal DFA for the language  $L(A \circ B)$  has  $\alpha$  states.*

*Proof.* We give a proof by contradiction. So we assume that  $\alpha$  is an integer in  $[2, nm]$ , which is coprime to  $n$ , and  $A$  is a  $n$ -state and  $B$  is a  $m$ -state PFA, such that the minimal DFA accepting  $L(A \circ B)$  has exactly  $\alpha$  states. Let  $Q_A$  be the state set of  $A$  and  $Q_B$  the state set of  $B$ , respectively.

By Theorem 4 we know, that there are  $nx$  states, for some  $1 \leq x \leq m$ , that are initially reachable in  $A \circ B$ , and that  $\alpha$  must divide  $nx$ . Therefore, it is possible to partition the set of initially reachable states of  $A \circ B$  into  $\alpha$  sets such that each set contains exactly  $nx/\alpha$  states which are equivalent. Let these sets be  $T_0, T_1, \dots, T_{\alpha-1}$ . Define

$$S_i := \{q \in Q_A \mid \text{there exists } p \in Q_B \text{ such that } (q, p) \in T_i\}.$$

Because  $A \circ B$  is a PFA, for every pair of states there exists a word which maps one of those states onto the other. Since every word acts directly on the first component of a state we find that  $|(\{q\} \times Q_B) \cap T_i|$  is equal for every state  $q$  of  $A$  for which  $S_i$  is not empty. Since it makes the further considerations a lot easier we fix  $q$  as an arbitrary state of  $A$ . Again, by Theorem 4 we know, that for every state  $q$  of  $A$  there are  $x$  states (with first component  $q$ ) initially reachable in  $A \circ B$ . Thus, we obtain for a set  $T_i$ , which contains a state that has  $q$  as its first component

$$|(\{q\} \times Q_B) \cap T_i \neq \emptyset \mid 0 \leq i \leq \alpha - 1| \cdot |(\{q\} \times Q_B) \cap T_i| = x,$$

and  $|(\{q\} \times Q_B) \cap T_i| \cdot |Q_A \cap S_i| = |T_i| = n \frac{x}{\alpha}$ . By inserting the first equation into the second we obtain

$$\begin{aligned} & |(\{q\} \times Q_B) \cap T_i| \cdot |Q_A \cap S_i| \\ &= n \cdot \frac{|(\{q\} \times Q_B) \cap T_i \neq \emptyset \mid 0 \leq i \leq \alpha - 1| \cdot |(\{q\} \times Q_B) \cap T_i|}{\alpha}. \end{aligned}$$



Dividing by  $|(\{q\} \times Q_B) \cap T_i|$  gives us

$$|Q_A \cap S_i| = n \cdot \frac{|\{(\{q\} \times Q_B) \cap T_i \neq \emptyset \mid 0 \leq i \leq \alpha - 1\}|}{\alpha}.$$

Since  $|Q_A \cap S_i|$  is an integer and the numbers  $n$  and  $\alpha$  are coprime we obtain that  $|\{(\{q\} \times Q_B) \cap T_i \neq \emptyset \mid 0 \leq i \leq \alpha - 1\}|$  is divisible by  $\alpha$ . This in turn implies that it is equal to  $\alpha$ , since it is upward limited by  $\alpha$ . Therefore, for every set  $T_i$ , for  $0 \leq i \leq \alpha - 1$ , there exists a state which has  $q$  as its first component. Because  $\alpha$  is at least two, there must be an initially reachable accepting state in  $A \circ B$ . Thus, there exists an  $i$  with  $0 \leq i \leq \alpha - 1$ , such that  $T_i$  contains only (equivalent) accepting states. In conclusion that means that  $q$  must be accepting. Since state  $q$  was arbitrarily chosen this implies that every state of  $A$  must be accepting, which is a contradiction to the fact that  $A$  is minimal.  $\square$

### 3.3 Cascade Products with Permutation and Permutation-Reset Automata and Beyond

Next we investigate the descriptive complexity of the cascade product of a permutation and a permutation-reset device. For the first case, where the first automaton of the cascade product is a PFA no magic numbers exist. This is in contrast to the unary case [6].

**Theorem 5.** *Let  $n, m \geq 2$ . Then for every  $\alpha$  with  $1 \leq \alpha \leq nm$ , there exists a minimal binary  $n$ -state PFA  $A$  and a minimal  $m$ -state PRFA  $B$  such that the minimal DFA accepting the language  $L(A \circ B)$  has exactly  $\alpha$  states.*

*Proof (Sketch).* In [6] it was shown that in the unary case the numbers  $[1, 2n]$  are reachable. Thus, we may assume that  $\alpha > 2n$ . One can show that every number  $\alpha$  in the interval  $[n, nm]$  can be written in the form  $\alpha = km + n - k + \ell$ , for integers  $0 \leq k \leq n - 1$  and  $0 \leq \ell \leq m - 1$ . In order to simplify the constructions to come, we want to exclude the case  $\ell = m - 1$ . In case  $\alpha$  has a representation as above with  $\ell = m - 1$ , we may rewrite it to  $\alpha = km + n - k + (m - 1) = (k + 1)m + n - (k + 1)$ , as long as  $k < n - 1$ . For  $k = n - 1$ , the value of  $\alpha$  is equal to  $nm$ , which can be reached by a result in [6] already in the unary case. In summary,  $\alpha$  belongs now to the interval  $[2n + 1, nm - 1]$  and can be written as above with  $0 \leq k \leq n - 1$  and  $0 \leq \ell < m - 1$ .

Let  $A = (\{q_0, q_1, \dots, q_{n-1}\}, \{a, b\}, \cdot_A, q_0, F_A)$ , with

$$\begin{aligned} q_i \cdot_A a &= q_{i+1 \bmod n}, & \text{for } 0 \leq i \leq n - 1, \\ q_i \cdot_A b &= q_i, & \text{for } 0 \leq i \leq n - k - 2, \\ q_i \cdot_A b &= q_{i+1}, & \text{for } n - k - 1 \leq i \leq n - 3, \\ q_{n-2} \cdot_A b &= q_{n-k-1}, \\ q_{n-1} \cdot_A b &= q_{n-1}, \end{aligned}$$

where all non-specified transitions are self-loops as usual, and  $F_A = \{q_{n-1}\}$ , if  $n = 2$  and  $k = 0$ ,  $F_A = \{q_{n-2}\}$ , if  $n = 2$  and  $k > 0$ , and  $F_A = \{q_{n-2}, q_{n-1}\}$ ,

otherwise. That  $A$  is minimal follows for  $n = 2$  by Lemma 1. In case  $n > 2$  the minimality of the device  $A$  is seen because the states  $q_{n-2}$  and  $q_{n-1}$  are distinguishable by applying the word  $a$  and for every other state pair there exists a bijection which maps at least one of them into  $\{q_{n-2}, q_{n-1}\}$ .

Next let the PRFA  $B$  be

$$B = (\{p_0, p_1, \dots, p_{m-1}\}, \{q_0, q_1, \dots, q_{n-1}\} \times \{a, b\}, \cdot_B, p_0, \{p_0\}),$$

where

$$p_i \cdot_B (q_{n-2}, a) = \begin{cases} p_{i+1 \bmod m}, & \text{for } 0 \leq i \leq m-1 \text{ and } k = 0, \\ p_i, & \text{for } 0 \leq i \leq m-1 \text{ and } k > 0, \end{cases}$$

$$p_i \cdot_B (q_{n-2}, b) = \begin{cases} p_i, & \text{for } 0 \leq i \leq m-1 \text{ and } k = 0, \\ p_{i+1 \bmod m}, & \text{for } 0 \leq i \leq m-1 \text{ and } k > 0, \end{cases}$$

$$p_i \cdot_B (q_{n-1}, a) = \begin{cases} p_i, & \text{for } 0 \leq i \leq m-1, n = 2 \text{ and } k > 0, \\ p_0, & \text{for } 0 \leq i \leq m-1, n > 2 \text{ or } k = 0, \end{cases}$$

and

$$p_i \cdot_B (q_{n-1}, b) = p_{i+1 \bmod \ell+1}, \quad \text{for } 0 \leq i \leq \ell,$$

and all non-specified transitions are self-loops. The given construction ensures that all states in  $B$  are reachable (by the sole letter  $(q_{n-2}, a)$ , if  $k = 0$ , and by the letter  $(q_{n-2}, b)$ , otherwise). Thus, Lemma 1 shows that the automaton is minimal in all cases.

For the analysis of the cascade automaton  $A \circ B$  we similarly proceed as in the proof of Theorem 2. First one shows that the following three sets

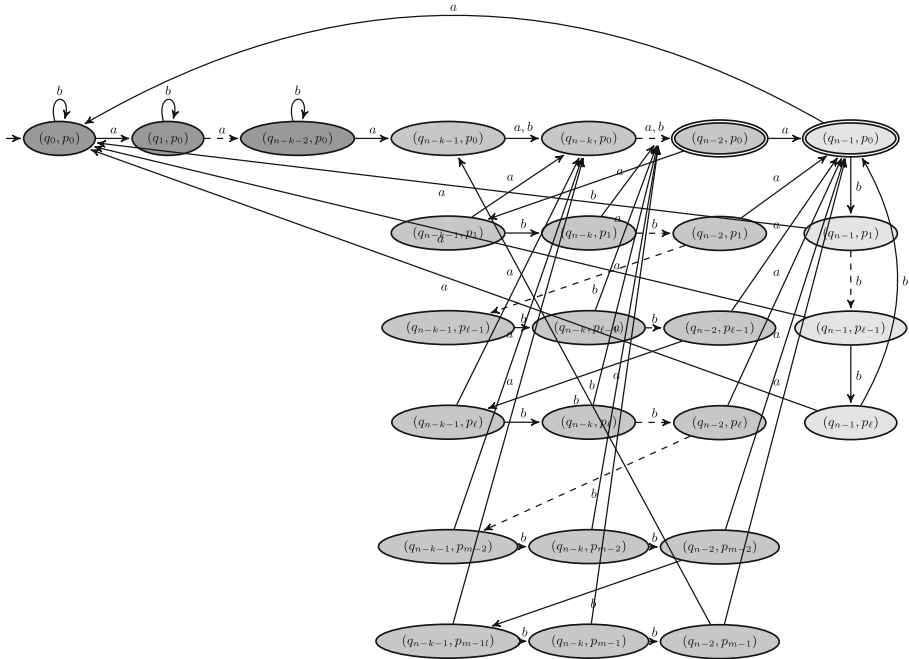
$$E_{a \leq n-k-2} = \{(q_0, p_0) \cdot a^j \mid 0 \leq j \leq n-k-2\},$$

$$E_{a^{n-k-1}b^*} = \begin{cases} \{(q_0, p_0) \cdot a^{n-k-1}b^i \mid i \geq 0\} & \text{if } k > 0, \\ \emptyset & \text{otherwise,} \end{cases}$$

and

$$E_{a^{n-1}b^*} = \{(q_0, p_0) \cdot a^{n-1}b^i \mid i \geq 0\},$$

where  $\cdot$  refers to the transition function of  $A \circ B$ , form the initially reachable states of  $A \circ B$ . To this end one distinguishes two cases depending on  $n$  with appropriate subcases on  $k$ ; the sets under consideration, e.g., for the case  $n > 2$  and  $k > 0$  are depicted in Fig. 4. In all the considered cases it turns out that  $A \circ B$  has  $\alpha$  initially reachable states. Minimality of the automaton  $A \circ B$  is shown by proving that the states in  $E_{a \leq n-k-2} \cup E_{a^{n-k-1}b^*} \cup E_{a^{n-1}b^*}$  are pairwise inequivalent. The cumbersome details are left to the reader. Then this proves the stated claim.  $\square$



**Fig. 4.** The initially reachable states of  $A \circ B$  in the case  $n > 2$  and  $k > 0$ . The states of  $E_{a \leq n-k-2}$ ,  $E_{a^{n-k-1}b^*}$ , and  $E_{a^{n-1}b^*}$  are coloured dark gray, gray, and light gray, respectively.

Obviously this theorem implies that the minimal DFA accepting  $L(A \circ B)$  for a  $n$ -state device  $A$  and a  $m$ -state device  $B$  can have every number of states in the integer interval  $[1, nm]$  in the following cases:

- $A$  is a binary PFA and  $B$  is an arbitrary DFA,
- $A$  is a binary PRFA and  $B$  is a PRFA,
- $A$  is an arbitrary binary DFA and  $B$  is a PRFA, and
- $A$  is an arbitrary binary DFA and  $B$  is an arbitrary DFA,

where both automata are always assumed to be minimal.

So it only remains to study the cascade product of an  $n$ -state PRFA  $A$  and an  $m$ -state PFA  $B$ . Recall that in the unary case magic numbers exist in this case [6].

**Theorem 6.** *Let  $n, m \geq 2$ . Then for every  $\alpha$  with  $1 \leq \alpha \leq nm$ , there exists a minimal binary  $n$ -state PRFA  $A$  and a minimal  $m$ -state PFA  $B$  such that the minimal DFA accepting the language  $L(A \circ B)$  has exactly  $\alpha$  states.*

Obviously, this statement generalizes to the remaining missing cases.

## 4 Conclusions

We continued our research on the magic number problem for the cascade product on finite automata of certain types. Compared to the unary case, which was studied in [6], where in almost all cases magic numbers were identified, here the situation is completely the other way around. In all cases, except one (Lemma 2), we do not find magic numbers. In fact, only for the cascade product of two minimal PFAs, a precise answer to the reachable states sizes are missing. Moreover, since the cascade product as introduced here uses final states, it also remains to study the effect on the descriptonal complexity of the choice of final states in the product automaton.

## References

1. Ae, T.: Direct or cascade product of pushdown automata. *J. Comput. Syst. Sci.* **14**(2), 257–263 (1977)
2. Arbib, M.A.: *Algebraic Theory of Machines, Languages, and Semigroups*. Academic Press, Cambridge (1968)
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](https://doi.org/10.1007/978-3-642-39310-5_26)
4. Čevorová, K., Jirásková, G., Krajňáková, I.: On the square of regular languages. In: Holzer, M., Kutrib, M. (eds.) CIAA 2014. LNCS, vol. 8587, pp. 136–147. Springer, Cham (2014). [https://doi.org/10.1007/978-3-319-08846-4\\_10](https://doi.org/10.1007/978-3-319-08846-4_10)
5. Harrison, M.A.: *Introduction to Formal Language Theory*. Addison-Wesley, Boston (1978)
6. Holzer, M., Rauch, C.: The range of state complexities of languages resulting from the cascade product—the unary case (extended abstract). In: Maneth, S. (eds.) *Implementation and Application of Automata*. CIAA 2021. Lecture Notes in Computer Science, vol. 12803. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-79121-6\\_8](https://doi.org/10.1007/978-3-030-79121-6_8)
7. Holzer, M., Hospodár, M.: The range of state complexities of languages resulting from the cut operation. In: Martín-Vide, C., Okhotin, A., Shapira, D. (eds.) LATA 2019. LNCS, vol. 11417, pp. 190–202. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-13435-8\\_14](https://doi.org/10.1007/978-3-030-13435-8_14)
8. Hricko, M., Jirásková, G., Szabari, A.: Union and intersection of regular languages and descriptonal complexity. In: Mereghetti, C., Palano, B., Pighizzini, G., Wotschke, D. (eds.) *Proceedings of the 7th Workshop on Descriptonal Complexity of Formal Systems*, pp. 170–181. Università degli Studi di Milano, Como (2005)
9. Iwama, K., Kambayashi, Y., Takaki, K.: Tight bounds on the number of states of DFAs that are equivalent to  $n$ -state NFAs. *Theor. Comput. Sci.* **237**(1–2), 485–494 (2000)
10. Jirásková, G.: Magic numbers and ternary alphabet. *Internat. J. Found. Comput. Sci.* **22**(2), 331–344 (2011)

11. Maler, O.: On the Krohn-Rhodes cascaded decomposition theorem. In: Manna, Z., Peled, D.A. (eds.) *Time for Verification*. LNCS, vol. 6200, pp. 260–278. Springer, Heidelberg (2010). [https://doi.org/10.1007/978-3-642-13754-9\\_12](https://doi.org/10.1007/978-3-642-13754-9_12)
12. Maler, O., Pnueli, A.: Tight bounds on the complexity of cascaded decomposition of automata. In: *Proceedings of the 31st Annual Symposium on Foundations of Computer Science*, pp. 672–682. IEEE Computer Society Press, St. Louis (1990)