



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

Markus Holzer^(✉) and Christian Rauch

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

Abstract. We investigate the state complexity of languages resulting from the cascade product of two minimal deterministic finite automata with n and m states, respectively. More precisely we study the magic number problem of the cascade product operation and show what range of complexities can be produced in case the left automaton is unary, that is, has only a singleton letter alphabet. Here we distinguish the cases when the involved automata are reset automata, permutation automata, permutation-reset automata, or do not have any restriction on their structure. It turns out that the picture on the obtained state complexities of the cascade product is diverse, and for all cases, except where the left automaton is a unary permutation(-reset) or a deterministic finite automaton without structural restrictions, and the right one is a reset automaton or a deterministic finite automaton without structural restrictions, we are able to identify state sizes that *cannot* be reached—these numbers are called “magic.”

1 Introduction

Originally, the magic number problem for finite automata [8] asks whether there exists a minimal n -state nondeterministic finite automaton whose equivalent minimal deterministic finite automaton (DFA) has α states for all n and α with $n \leq \alpha \leq 2^n$. A number α not satisfying this condition is called a *magic number* for n . The problem was solved in [9], showing that for ternary languages *no* magic numbers exist contrary to the unary case [3]. It is worth noting that for binary languages the original problem from [8] is still open.

Observe that the idea behind the magic number problem is not limited to the determinization of nondeterministic finite automata. In fact, shortly after the introduction of the magic number problem several papers studied regularity preserving operations from a magic number perspective. In [7] it was shown that for the intersection of DFAs *no* number in the interval $[1, nm]$ is magic—this already holds for binary automata. Besides intersection also other formal language operations such as, e.g., union [7], concatenation [9], square [2], star [1],

reversal, and the cut operation [6] were investigated on the quest for magic numbers. It turned out that magic numbers are quite rare. For instance, for the star of unary languages there are linearly many magic numbers [1]. On the other hand, star of binary languages has no magic numbers. For the cut operation on unary automata the interval $2m$ up to $n - 1$ turns out to be magic. Thus, these complexities cannot be reached by the cut operation on m - and n -state DFAs, if $2m \leq n - 1$.

We contribute to the list of magic number problems for operations on automata by studying the cascade product, which is the main ingredient to the celebrated Krohn-Rhodes Theorem [10] that states that any finite automaton can be decomposed into (several) simple “automata prime factors.” Here we are not interested in the classification of regular languages by automata prime factors. Instead, we investigate the descriptive complexity of the cascade operation on two finite automata A and B , only. We further limit our study to the case where the left automaton A is unary. For a better fine grain investigation of the subject in question we use minimal DFAs only from the following automata classes as operands to the cascade product operation: reset automata (RFA), permutation automata (PFA), permutation-reset automata (PRFA), and automata with no structural restrictions (DFA)—for unary automata every permutation-reset automaton is in fact either a reset or a permutation automaton. We list our findings in Table 1. A careful inspection of the table reveals that the right automaton B in the cascade product is more important for the number of reachable states in the product than the left one. Moreover, the picture turns out to be quite diverse. For instance, in case the left automaton is a unary RFA all combinations for the cascade product lead to magic numbers, which is not the case for the remaining products where a RFA is the right automaton in the product. The most complex situation appears whenever PFAs or PRFAs are involved. In these cases the set structure of the reachable number of states is mostly determined by the size n of the left automaton and the non-trivial divisors t of n . These cases lead to a significant amount of magic numbers. In fact, for all cases where magic numbers exist, except for the case where both devices are RFAs, the number $nm - 1$ turns out to be always magic. For the cascade product of an n -state unary PFA or a DFA with an m -state finite automaton in general the whole range $\{1, 2, \dots, nm\}$ can be obtained, and thus no magic numbers exist in these cases. This is not too surprising since the cascade product can simulate the intersection of two automata—compare with [7]. The obtained results are in sharp contrast to the general case, when we do not restrict to unary automata as left operands in the cascade product. In [5] it was shown that for the general case, magic numbers only exist for the cascade product of two permutation automata. In all other cases the cascade products do not have magic numbers at all.

2 Preliminaries

We recall some definitions on finite automata as contained in [4]. A *deterministic finite automaton* (DFA) is a quintuple $A = (Q, \Sigma, \cdot, q_0, F)$, where Q is the finite

Table 1. The range of state complexities for the cascade product of a minimal unary n -state automaton A and a minimal m -state finite state device B of the mentioned types. The parameter t used in the set descriptions is a non-trivial divisor of n or k , depending on the case we are in. Moreover, the operation \oplus on sets of numbers S_1 and S_2 is defined as $S_1 \oplus S_2 = \{x + y \mid x \in S_1 \text{ and } y \in S_2\}$. In all cases where magic numbers exist, except for the cascade product of two RFAs, the number $nm - 1$ turns out to be magic.

Automata		State complexities of minimal DFAs for $L(A \circ B)$	Magic number(s)	
A	B			
RFA	RFA	$\{1, 2, 3\}$	Yes	Theorem 5
	PFA	$\{1, 2, \dots, m + 1\}$	Yes	Theorem 6
	PRFA			
	DFA			
PFA	RFA	$\{1, 2, \dots, 2n\}$	No	Theorem 7
	PFA	$\{1\} \cup \{nx \mid 1 \leq x \leq m\} \cup \{tx \mid 1 \leq x < m\}$	Yes	Theorem 9
	PRFA	$\{1, 2, \dots, 2n\} \cup \{nx \mid 1 \leq x \leq m\} \cup \{tx \mid 1 \leq x < m\}$	Yes	Theorem 17
	DFA	$\{1, 2, \dots, nm\}$	No	Theorem 18
DFA	RFA	$\{1, 2, \dots, 2n\}$	No	Theorem 7
	PFA	$\bigcup_{k=1}^n (M_k \oplus [0, n - k])$, where	Yes	Theorems 19 and 20
	PRFA	$M_k = \{1\} \cup \{kx \mid 1 \leq x \leq m\} \cup \{tx \mid 1 \leq x < m\}$		
	DFA	$\{1, 2, \dots, nm\}$	No	Theorem 18

set of *states*, Σ 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 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 on 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 Σ 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 the following chain of inclusions $XFA \subseteq PRFA \subseteq FA$, where $X \in \{P, R\}$, holds. Moreover, it is not hard to see that the classes RFA and PFA are incomparable.

In [10] 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

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

for $p \in Q_A$, $q \in Q_B$, and $a \in \Sigma$. We say that A is the *left automaton* and B the *right automaton* in the cascade product $A \circ B$. It is obvious that the cascade product of two DFAs contains their direct product. In order to explain our notation we give a small example.

Example 1. Consider the PFA $A = (\{q_0, q_1, q_2, q_3\}, \{a\}, \cdot_A, q_0, \{q_0, q_1, q_2\})$, where the transitions are given by $q_i \cdot_A a = q_{i+1 \bmod 4}$, for $0 \leq i \leq 3$. Moreover, let

$$B = (\{p_0, p_1\}, \{q_0, q_1, q_2, q_3\} \times \{a\}, \cdot_B, p_0, \{p_0\})$$

be the PFA, where for all states and letters the transition function \cdot_B acts like the identity, except for the letters (q_0, a) and (q_1, a) . In this case, let $p_0 \cdot_B (q_0, a) = p_1$ and $p_1 \cdot_B (q_0, a) = p_0$. Moreover, let $p_0 \cdot_B (q_1, a) = p_1$ and $p_1 \cdot_B (q_1, a) = p_0$. The automata A and B are depicted in Fig. 1 on the top and lower right, respectively. It is easy to see that both automata are minimal.

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

$$A \circ B = (\{q_0, q_1, q_2, q_3\} \times \{p_0, p_1\}, \Sigma, \cdot, (q_0, p_0), \{q_0, q_1, q_2\} \times \{p_0\}),$$

where the transition function can be deduced from Fig. 1 on the lower left. Observe, that $A \circ B$ is also a PFA and that not all states are initially reachable. From the initially reachable part of $A \circ B$ the states (q_0, p_0) and (q_2, p_0) (states (q_1, p_1) and (q_3, p_0) , respectively) are equivalent. Because these states are the only initially reachable ones and only two of the four are accepting, the minimal DFA which accepts the language $L(A \circ B)$ has exactly two states.

The following result is immediate by the lower bound results on the operational complexity of the intersection operation on finite automata [7].

Theorem 2. *Let A be an n -state and B an m -state DFA. Then nm 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 descriptonal 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 right 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. If the left 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 right automaton accepts under the bijective mapping $(q, a) \mapsto a$ for the letters a of the left automaton, where q is the state of the left automaton. Therefore, only 1- or m -state automata, for $m \geq 1$, appear as results of a cascade product with a trivial automaton. Thus, in the forthcoming we only consider non-trivial automata.

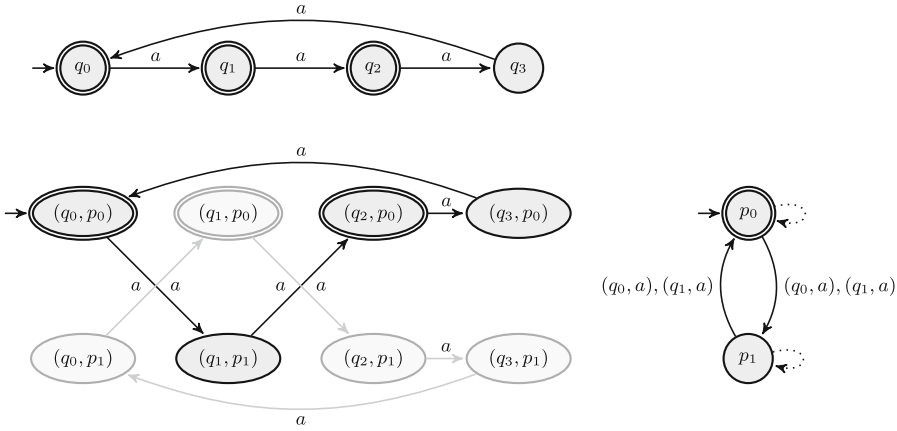


Fig. 1. The example automata A and B on the top and lower right, respectively. For a better visibility not all transitions of an automaton are shown. In particular, this is the case for automaton B , where self-loops are only depicted by dotted loops without letters. The cascade product $A \circ B$ is depicted on the lower left.

3 Results on the Cascade Product

This section is fourfold. In the first subsection we investigate the magic number problem for the cascade product, where at least one automaton is a reset device, while in the second subsection we study the magic number problem when both automata are PFAs. Afterwards we study the case, where the left automaton is a PFA and the right automaton is a PRFA. Finally we investigate the magic number problem for the cascade product, where at least one automaton is an arbitrary DFA. Before we start our studies we present a lemma on the minimality of PFAs that is used very often in the subsections to come without further notice. It is helpful and provides important information about the properties of PFAs.

Lemma 3. *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.*

Now we are ready for the first subsection considering the cascade product, where at least one automaton is a reset device.

3.1 At Least One Automaton is a Reset Automaton

Before we start our investigation on the cascade product where at least one automaton is a RFA, we take a closer look on minimal reset devices. It is easy to see that one cannot distinguish more than two non-accepting states, because the word that proves both of these states distinguishable must contain at least one letter that acts as a reset and therefore after reading this letter both states are mapped to the same state and thus cannot be shown inequivalent anymore.

A similar reasoning applies to accepting states. Hence, every minimal RFA has at most one accepting and one non-accepting state. Thus, we have shown the following result, where the single state case is trivial.

Lemma 4. *Every minimal reset automaton has at most two states.* \square

Although minimal RFAs form a very restricted class of automata their cascade product is worth to be considered in detail. We find the following situation—recall, that we only deal with non-trivial automata:

Theorem 5. *Let A and B be two minimal non-trivial RFAs, that is, both devices are 2-state automata. (i) If A is a unary automaton, then the minimal DFA accepting the language $L(A \circ B)$ has α states with $1 \leq \alpha \leq 3$ and (ii) if A has an input alphabet of at least two letters, then the minimal DFA accepting the language $L(A \circ B)$ has α states with $1 \leq \alpha \leq 4$.*

One can try to generalize the results of the previous theorem to other automata classes such as permutation automata for the right automaton in the product. In fact, one can show that for a non-trivial minimal unary RFA A and a minimal m -state PFA B , any DFA accepting the language $A \circ B$ has at most $m + 1$ states. Next we show that the whole interval $[1, m + 1]$ can be reached, if the left automaton is a minimal non-trivial unary RFA.

Theorem 6. *Let $m \geq 1$. Then for every α with $1 \leq \alpha \leq m + 1$, there exists a minimal non-trivial unary RFA A and a minimal m -state PFA B such that the minimal DFA for the language $L(A \circ B)$ has α states.*

This completes the case where the right automaton of the cascade product is a PFA. Now the question arises what happens if the PFA appears as the left automaton in the cascade product with a RFA. Compared to the previous case already for minimal unary PFAs and non-trivial RFAs the whole interval $[1, 2n]$, where n is the number of states of the PFA, can be reached. Observe that in the unary case the next theorem is in stark contrast to Theorem 6.

Theorem 7. *Let $n \geq 1$. Then for every α with $1 \leq \alpha \leq 2n$, there exists a minimal unary n -state PFA A and a minimal non-trivial RFA B such that the minimal DFA accepting the language $L(A \circ B)$ has exactly α states. The result holds true even in the case if A has an input alphabet of arbitrary size.* \square

Since $\text{PFA} \subseteq \text{PRFA} \subseteq \text{FA}$ holds the results from this subsection, where PFAs are involved, immediately generalize to permutation-reset and finite automata in general.

3.2 Two Permutation Automata

Before we start with the descriptive complexity analysis of the cascade product of two permutation automata we prove a useful result that is helpful to determine which deterministic state complexities are reachable and which ones are not.

Lemma 8. *The cascade product $A \circ B$ of two permutation automata A and B is a permutation automaton, too.*

Now we are ready to give an overview of all possible deterministic state complexities that can arise. We call a divisor t of a number n *non-trivial* if t is neither equal to 1 or n .

Theorem 9. *Let $n, m \geq 2$ and t be a non-trivial divisor of n . Then for every α in $\{1\} \cup \{nx \mid 1 \leq x \leq m\} \cup \{tx \mid 1 \leq x < m\}$, and only for those, there exists a minimal unary n -state PFA A and a minimal m -state PFA B such that the minimal DFA for the language $L(A \circ B)$ has α states.*

In a series of lemmata we first show how to reach each of the above specified values. Afterwards we show that only these values can be obtained. We start with the values of the form nx , for $1 \leq x \leq m$.

Lemma 10. *Let $n, m \geq 2$ and x with $1 \leq x \leq m$. Then for every α that is equal to nx , there exists a minimal unary n -state PFA A and a minimal m -state PFA B such that the minimal DFA for the language $L(A \circ B)$ has α states.*

With additional effort we can also show that every divisor of n is also reachable. This obviously includes the cases 1 and t of Theorem 9.

Lemma 11. *Let $n, m \geq 2$. Then for every α that is equal to one or to a non-trivial divisor of n , there exists a minimal unary n -state PFA A and a minimal m -state PFA B such that the minimal DFA for the language $L(A \circ B)$ has α states.*

We can extend the statement from the above lemma to the multiples of the divisors of n with some side conditions.

Lemma 12. *Let $n, m \geq 2$. Moreover, assume that x satisfies $2 \leq x \leq m-1$ and that t is a non-trivial divisor of n . Then for every α that is equal to tx , there exists a minimal unary n -state PFA A and a minimal m -state PFA B such that the minimal DFA for the language $L(A \circ B)$ has α states.*

The Lemmata 10, 11, and 12 thus show the reachability of the number of states in the cascade product of a unary PFA with a PFA as claimed in Theorem 9. Hence, it remains to prove that these are the only numbers that can be obtained. To this end we first prove two structural properties of cascade products of PFAs.

Lemma 13. *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.*

The next lemma provides information about the equivalence classes of the cascade product of permutation automata.

Lemma 14. *Let $A \circ B$ be the cascade product of two minimal PFAs A and B . Then the minimal deterministic finite automata that accepts $L(A \circ B)$ has α states, where α is a divisor of the quantity of initially reachable states of $A \circ B$. Furthermore, every state of $A \circ B$ has the same number of equivalent states, if $A \circ B$ is strongly connected.*

The last two lemmata obviously imply that only the numbers in

$$\{tx \mid t \text{ is a divisor of } n \text{ and } 1 \leq x \leq m\}$$

can be reached in the cascade product of two PFAs. This set differently written is equal to

$$\begin{aligned} &\{1, 2, \dots, m\} \cup \{nx \mid 1 \leq x \leq m\} \\ &\cup \{tx \mid t \text{ is a non-trivial divisor of } n \text{ and } 1 \leq x < m\} \\ &\cup \{tm \mid t \text{ is a non-trivial divisor of } n\}, \end{aligned}$$

where the unions are eventually *not* disjoint. In order to prove Theorem 9 it remains to exclude those numbers α that do not have a representation as given there. Because we showed already that $\alpha = 1$ is reachable we assume that $\alpha \geq 2$. Due to the Lemma 13 we know that the number of initially reachable states in $A \circ B$ is nx , for an integer $1 \leq x \leq m$. Moreover, by Lemma 14 we know that α is a divisor of nx . Now we distinguish two cases depending on the greatest common divisor t of n and α .

1. Case $t = 1$. Recall that α is the number of states of the minimal automaton accepting $L(A \circ B)$. First observe that the word a^α is the shortest word that only permutes equivalent states of $A \circ B$ and on the other hand the word a^n is the shortest word which induces the identity mapping on the states of A . Because α and n are coprime the smallest word which fulfills both conditions is $a^{n\alpha}$. This in turn implies that every mapping $a^{j\alpha}$, for $1 \leq j \leq n$, has a different image in A for a given state. Because $\alpha \geq 2$, there is at least one accepting and one non-accepting state that is initially reachable in $A \circ B$. We pick an arbitrary initially reachable accepting state (q, p) in $A \circ B$. Then by applying the mappings $a^{j\alpha}$, for $1 \leq j \leq n$, to (q, p) one observes that every of the obtained images has a different first component. Because (q, p) is accepting we obtain that n different states of A have to be accepting, which is a contradiction to the minimality of A .
2. Case $t > 1$. Then we distinguish two subcases:
 - (a) Assume $\alpha/t \neq m$. Trivially, α equals $t \cdot \alpha/t$, where t is a divisor of n and α/t is a divisor of x . Because $t > 1$ we obtain the reachability of α by the Lemmata 10 and 12.
 - (b) In this case we observe that $\alpha = t\alpha/t = tm$ and because there is no other common divisor of n and α it follows that n/t and m are coprime. We will show in the following theorem that α is not reachable in this case.

Theorem 15. *Let $n, m \geq 2$ and t be a non-trivial divisor of n . Then for every α that is equal to tm , there does not exist a minimal unary n -state PFA A and a minimal m -state PFA B such that the minimal DFA for the language $L(A \circ B)$ has α states, if the numbers m and $\frac{n}{t}$ are relatively prime.*

This completes our investigation on the cascade product of two permutation automata and eventually proves Theorem 9. Finally we want to point out that for example the numbers $nm - 1$ are magic numbers for every non-trivial minimal n -state PFA A and minimal m -state PFA B . This can be easily seen because for $n, m \geq 2$ we have (i) $1 < 3 \leq nm - 1$, (ii) $n(m - 1) < nm - 1 < nm$, and (iii) $tx \leq t(m - 1) < nm - 1$, for every non-trivial divisor t of n and $1 \leq x \leq m - 1$. Therefore the reachability of $nm - 1$ is excluded by Theorem 9.

3.3 Permutation Automata with Permutation-Reset Automata

The next case that we consider for the cascade product is that of a unary permutation automaton with a permutation-reset device. We will see that a few further numbers on the state complexity are added to the case considered in the previous subsection. We start with the following lemma.

Lemma 16. *Let $n, m \geq 2$. Then for every α with $1 \leq \alpha \leq 2n$, there exists a minimal unary n -state PFA A and a minimal non-trivial PRFA B such that the minimal DFA accepting the language $L(A \circ B)$ has exactly α states.*

Since permutation-reset automata subsume permutation and reset automata we may safely conclude that at least all state sizes that appear in the cascade product $A \circ B$ of a permutation automaton A with an automaton B of the above types can be reached. Thus, by Theorems 7, 9, and Lemma 16 this results in the set $\{1, 2, \dots, 2n\} \cup \{nx \mid 1 \leq x \leq m\} \cup \{tx \mid 1 \leq x < m\}$ of reachable state numbers. In the following theorem we show that these are indeed the only cases that can be reached for the cascade product of a unary PFA with a PRFA.

Theorem 17. *Let $n, m \geq 2$ and t be a non-trivial divisor of n . Then for every α in $\{1, 2, \dots, 2n\} \cup \{nx \mid 1 \leq x \leq m\} \cup \{tx \mid 1 \leq x < m\}$, and only for those, there exists a minimal unary n -state PFA A and a minimal m -state PRFA B such that the minimal DFA for the language $L(A \circ B)$ has α states.*

Clearly for $m = 2$ there are no magic numbers for the cascade product of a minimal n -state PFA and a minimal m -state PRFA. But for $m > 2$ we have that $2n < nm - 1$, and therefore $nm - 1$ is a magic number for every pair n, m with $n \geq 2$ and $m \geq 3$.

3.4 Deterministic Finite Automata Without Restrictions

In order to complete our study on the cascade product for unary automata it remains to consider the cases, where in particular the right automaton is allowed to be a DFA in general. We will show that in this case there do not exist magic numbers, i.e., we obtain the whole interval $[1, nm]$.

Theorem 18. *Let $n, m \geq 2$. Then for every α in the interval $[1, nm]$ there exists a minimal unary n -state PFA A and a minimal m -state DFA B such that the minimal DFA for the language $L(A \circ B)$ has α states.*

By Theorem 18 we know that there are no magic numbers in the integer interval $[1, nm]$, if we allow the right automaton to be an arbitrary DFA. Because $\text{PFA} \subset \text{PRFA} \subset \text{FA}$ we can transfer this result one-to-one to the case where both automata are DFAs.

It remains to consider the case where the left automaton is a DFA and the right one is a PFA or a PRFA. We need some notation for the next theorem: for two sets S_1 and S_2 of numbers let $S_1 \oplus S_2 := \{x + y \mid x \in S_1 \text{ and } y \in S_2\}$. Now we are ready for the statement, where the right automaton is a PFA.

Theorem 19. *Let $n, m \geq 2$. For k with $1 \leq k \leq n$ we define*

$$M_k = \{1\} \cup \{kx \mid 1 \leq x \leq m\} \\ \cup \{tx \mid t \text{ is a non-trivial divisor of } k \text{ and } 1 \leq x < m\}.$$

Observe that $M_1 = \{x \mid 1 \leq x \leq m\}$, because 1 does not have any non-trivial divisors. Then for every α in $\bigcup_{k=1}^n (M_k \oplus [0, n - k])$, and only for those, there exists a minimal unary n -state DFA A and a minimal m -state PFA B such that the minimal DFA for the language $L(A \circ B)$ has α states.

Finally, we show that there is no improvement on the reachable numbers if we use a PRFA B instead of a PFA as right operand in the cascade product with a DFA as left operand.

Theorem 20. *Let $n, m \geq 2$. Let M_k , for $1 \leq k \leq n$, be defined as in the previous theorem. Then for every α in $\bigcup_{k=1}^n (M_k \oplus [0, n - k])$, and only for those, there exists a minimal unary n -state DFA A and a minimal m -state PRFA B such that the minimal DFA for the language $L(A \circ B)$ has α states.*

One may ask whether all numbers in the integer interval $[1, nm]$ are reachable by Theorems 19 and 20. This is in fact not the case. For instance, if $n = 3$ and $m = 4$ then the reader may verify that we can only reach the values $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 12\}$, because

$$M_1 \oplus [0, n - 1] = (\{1, 2, 3, 4\} \oplus [0, 2]) = \{1, 2, 3, 4, 5, 6\}, \\ M_2 \oplus [0, n - 2] = (\{1, 2, 4, 6, 8\} \oplus [0, 1]) = \{1, 2, 3, 4, 5, 6, 7, 8, 9\},$$

and

$$M_3 \oplus [0, n - 3] = (\{1, 3, 6, 9, 12\} \oplus [0]) = \{1, 3, 6, 9, 12\}.$$

A list of all magic numbers, for n and m with $2 \leq n, m \leq 6$ is given in Table 2. The interested reader may have noticed that the number $nm - 1$ appears in all non-empty sets in the presented table. This holds in general and can be seen as follows: (i) the largest number describable by an addition of $n - k$ to the elements of M_k , for $k < n - 1$, is less or equal to $(n - 1)m + 1$, (ii) the largest number

in M_{n-1} is $(n - 1)m$, which gives the number $(n - 1)m + 1$, that for $m > 2$, is strictly less than $(n - 1)m + m - 1 = nm - 1$, (iii) the second largest number in the set M_n is $n(m - 1)$, which is strictly less than $nm - 1$, and (iv) the largest number in M_n is nm , which is strictly greater than $nm - 1$. This shows that $nm - 1$ is not a member of $\bigcup_{k=1}^n (M_k \oplus [0, n - k])$ and thus is a magic number.

Table 2. The sets of magic numbers for the cascade product of a minimal unary n -state DFA A and a minimal m -state PFA B , for $2 \leq n, m \leq 6$, w.r.t. the interval $[1, nm]$.

DFA A n -states	PFA or PRFA B with m states				
	2	3	4	5	6
2	\emptyset	{5}	{7}	{7, 9}	{9, 11}
3	\emptyset	{8}	{10, 11}	{13, 14}	{14, 16, 17}
4	\emptyset	{11}	{11, 14, 15}	{14, 17, 18, 19}	{17, 21, 22, 23}
5	\emptyset	{14}	{18, 19}	{18, 19, 22, 23, 24}	{22, 23, 26, 27, 28, 29}
6	\emptyset	{17}	{19, 22, 23}	{19, 23, 27, 28, 29}	{23, 27, 28, 29, 32, 33, 34, 35}

4 Conclusions

The Krohn-Rhodes Theorem [10] states that for every DFA A there exists a cascade product of PRFAs that is equivalent to A . The descriptonal complexity version of this statement [11, 12] gives exponential upper and lower bounds on the size of the cascade product of A . To our knowledge the descriptonal complexity of the cascade product for two automata was not investigated so far. We close this gap in this paper, by studying the problem in question for unary automata as left operands in the cascade product. In this way we are able to draw a complete picture for the studied cases and identify magic numbers, that is, size values that cannot be obtained by a cascade product of two minimal automata. See Table 1 for the obtained results in detail. The general problem, i.e., the descriptonal complexity of the cascade product for non-unary left operands is studied in [5].

References

1. 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
2. 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
3. Geffert, V.: Magic numbers in the state hierarchy of finite automata. *Inf. Comput.* **205**(11), 1652–1670 (2007)
4. Harrison, M.A.: Introduction to Formal Language Theory. Addison-Wesley (1978)

5. Holzer, M., Rauch, C.: The range of state complexities of languages resulting from the cascade product-the general case (2021). Proc. DLT, to appear
6. 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
7. Hricko, M., Jirásková, G., Szabari, A.: Union and intersection of regular languages and descriptional complexity. In: Mereghetti, C., Palano, B., Pighizzini, G., Wotschke, D. (eds.) Proceedings of the 7th Workshop on Descriptional Complexity of Formal Systems, pp. 170–181. Università degli Studi di Milano, Como, Italy (2005)
8. 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)
9. Jirásková, G.: Magic numbers and ternary alphabet. Int. J. Found. Comput. Sci. **22**(2), 331–344 (2011)
10. Krohn, K., Rhodes, J.: Algebraic theory of machines. Int. Trans. AMS **116**, 450–464 (1965)
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
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, Missouri, USA (1990)