IFIG

RESEARCH

REPORT

INSTITUT FÜR INFORMATIK

ON THE MAGIC NUMBER PROBLEM OF THE CUT OPERATION

Markus Holzer   Michal Hospodár

IFIG RESEARCH REPORT 1703

OCTOBER 2017

Institut für Informatik
JLU Gießen
Arndtstraße 2
35392 Giessen, Germany
Tel: +49-641-99-32141
Fax: +49-641-99-32149
mail@informatik.uni-giessen.de
www.informatik.uni-giessen.de

JUSTUS-LIEBIG-UNIVERSITÄT GIESSEN

IFIG Research Report
IFIG Research Report 1703, October 2017

On the Magic Number Problem of the Cut Operation

Markus Holzer[1]

Institut für Informatik, Universität Giessen
Arndtstraße 2, 35392 Giessen, Germany

and

Michal Hospodár[2]

Mathematical Institute, Slovak Academy of Sciences
Grešákova 6, 040 01 Košice, Slovakia

Abstract. We investigate the state complexity of languages resulting from the cut operation of two regular languages represented by deterministic finite automata with m and n states, respectively. We study the magic number problem of the cut operation and show that the entire range of complexities, up to the known upper bound, can be produced in case of binary alphabets. Moreover, we prove that in the unary case *only* complexities up to $2m-1$ and between n and $m+n-2$ can be produced, while if $2m \leq n-1$, then complexities within the interval $2m$ up to $n-1$ *cannot* be reached—these non-producible numbers are called "magic."

Categories and Subject Descriptors: F.1.1 [**Computation by Abstract Devices**]: Models of Computation—*Automata*; F.4.3 [**Mathematical Logic and Formal Languages**]: Formal Languages—*Classes defined by grammars or automata*; F.4.3 [**Mathematical Logic and Formal Languages**]: Formal Languages—*Operations on Languages*

Additional Key Words and Phrases: finite automata, cut operation, descriptional complexity, magic number problem, unary languages

[1] E-mail: holzer@informatik.uni-giessen.de
[2] E-mail: hosmich@gmail.com. Research supported by VEGA grant 2/0084/15 and grant APVV-15-0091.

1 Introduction

It is well known that for every n-state nondeterministic finite automaton one can always construct a language equivalent deterministic finite state device with at most 2^n states [15]. This bound is tight in the sense that for an arbitrary n there is always some n-state nondeterministic finite automaton which cannot be simulated by any deterministic finite state device with less than 2^n states [12, 13]. Nearly two decades ago a very fundamental question was raised in [9]: does there always exists a minimal n-state nondeterministic finite automaton whose equivalent minimal deterministic finite automaton 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. This simple question turned to be harder than expected. In a series of papers non-magic numbers were identified until the problem was solved in [10], showing that for ternary languages *no* magic numbers exist; for binary languages the original problem from [9] is still open. Contrary to the ternary case magic numbers do exist for unary languages as shown in [6]. A brief historical summary of the magic number problem can be found there.

The idea behind the magic number problem is not limited to the determinization of nondeterministic finite automata. In fact every (regularity preserving) formal language operation can be used to define a magic number problem for the operation in question. For instance, consider the intersection operation on languages. Let A and B be an m-state and n-state finite automaton, respectively. Then the lower and upper bound the output complexity for the intersection operation is 1 and mn. The former is induced by the empty set and the latter one by the standard cross-product construction for the intersection operation. Thus, in a similar way as for the determinization one may now ask, whether every α within the range between 1 and mn can be reached by intersecting two given automata with m and n states, respectively? In other words, is the outcome of the intersection operation in terms of the number of states continuous or are there any gaps, hence magic numbers? In [8] it was shown that for the intersection on deterministic finite automata *no* number in the interval $[1, mn]$ is magic—this already holds for binary automata. Besides intersection also other formal language operation such as, e.g., union [8], concatenation, square [2], star [3], and reversal were investigated from the "magic number" perspective. It turned out that magic numbers are quite rare. For instance, for the star of unary languages there are linearly many magic numbers [3]. On the other hand, star of binary languages has no magic numbers.

We contribute to the list of magic number problems for formal language operations by studying the cut operation, which was introduced in [1]. Roughly speaking, the cut operation is the machine implementation of concatenation on Unix text processors simulating the behavior of leftmost maximal matching. Tight upper bounds for the cut and iterated cut operation on deterministic finite automata were obtained in [5]. While the state complexity of concatenation is growing linearly with first parameter of the left automaton and exponentially with second parameter of the right automaton, the state complexity of the cut operation is only linearly growing with both parameters. We show that the entire range of complexities, up to the known upper bound, can be produced by the cut operation on deterministic finite automata with m and n states, respectively, in case of binary input alphabets. The proof of this result resembles some ideas used in [8] for the magic number problem on the union operation of deterministic finite automata. On the other hand, when restricting to unary automata we show that only complexities up to $2m - 1$ and between n and $m + n - 2$ can be reached, while complexities within the interval $2m$ up to $n - 1$ turn out to be magic. Thus, these complexities cannot be reached by the cut operation on m- and n-state deterministic finite automata, if $2m \leq n - 1$.

Here, as usual, the tail-loop structure of unary deterministic finite automata turns out to be very valuable in the proofs.

2 Preliminaries

We recall some definitions on finite automata as contained in [7]. Let Σ^* denote the set of all words over the finite alphabet Σ. The *empty word* is denoted by λ. Further, we denote the set $\{i, i+1, \ldots, j\}$ by $[i, j]$, if i and j are integers. A *deterministic finite automaton* (DFA) is a quintuple $A = (Q, \Sigma, \delta, s, F)$, 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 $\delta\colon Q \times \Sigma \to Q$ is the *transition function*. The *language accepted* by the DFA A is defined as

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

where the transition function is recursively extended to $\delta\colon Q \times \Sigma^* \to Q$. Two DFAs A and B are equivalent if they accept the same language, that is $L(A) = L(B)$. An automaton is *minimal* if it admits no smaller equivalent automaton w.r.t. the number of states. For DFAs this property can be easily verified by showing that all states are reachable from the initial state and all states are pairwise inequivalent.
In [1] the *cut operation* on languages K and L, denoted by $K \,!\, L$, is defined as

$$K \,!\, L = \{\, uv \mid u \in K,\ v \in L, \text{ and } uv' \notin K \text{ for every } v' \in \mathrm{pref}(v) \,\},$$

where $\mathrm{pref}(v)$ denotes the set of all nonempty prefixes of the word v. The above defined cut operation preserves regularity as shown in [1]. Since we are interested in the descriptional complexity of this operation we briefly recall the construction of a DFA for the cut operation; we slightly deviate from the presentation of the construction given in [5].

Let $A = (Q_A, \Sigma, \delta_A, s_A, F_A)$ and $B = (Q_B, \Sigma, \delta_B, s_B, F_B)$ be two DFAs accepting the languages K and L, respectively. The automaton that accepts the cut of K and L is drawn in a a row-column like fashion. To this end the DFA A is drawn in a column (its states are rows), and the DFA B is drawn in a row (its states are columns), see Figures 1 to 3. Then define the automaton $C = (Q, \Sigma, \delta, s, F)$, with state set $Q = (Q_A \times \{\bot\}) \cup (Q_A \times Q_B)$, where $\bot \notin Q_B$,

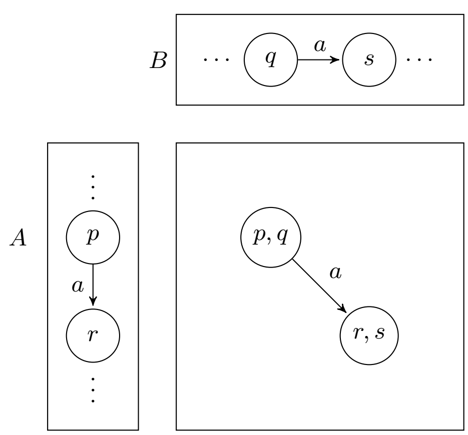

Fig. 1. The DFAs A and B and the DFA for $L(A) \,!\, L(B)$. We have $r = \delta_A(p, a)$ and $s = \delta_B(q, a)$. Notice that the state r is non-final.

and for each state (p, q) in Q and each input a in Σ we have

$$\delta((p,\perp),a) = \begin{cases} (\delta_A(p,a),\perp), & \text{if } \delta_A(p,a) \notin F_A; \\ (\delta_A(p,a),s_B), & \text{otherwise;} \end{cases}$$

and

$$\delta((p,q),a) = \begin{cases} (\delta_A(p,a),\delta_B(q,a)), & \text{if } \delta_A(p,a) \notin F_A, \text{ see Figure 1;} \\ (\delta_A(p,a),s_B), & \text{otherwise,} \qquad\qquad \text{see Figure 2;} \end{cases}$$

and $s = (s_A, \perp)$, if $\lambda \notin L(A)$, and $s = (s_A, s_B)$, otherwise. The set of final states is set to $F = Q_A \times F_B$, see Figure 3. Then $L(C) = K\,!\,L$.

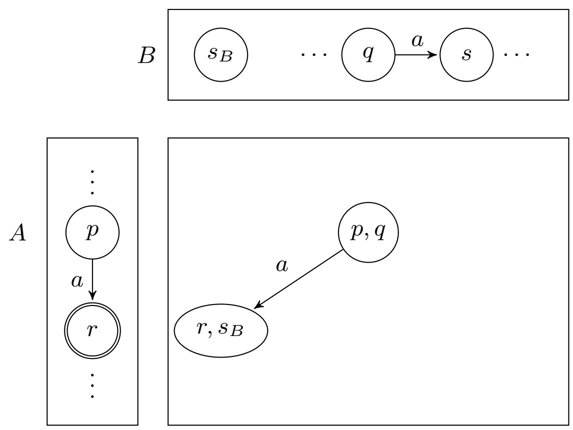

Fig. 2. The DFAs A and B and the DFA for $L(A)\,!\,L(B)$. We have $r = \delta_A(p,a)$ and $s = \delta_B(q,a)$. Notice that the state r is final.

We prove two useful results on the cut operation, which we use later during our investigations on the magic number problem of the operation in question. The results deal with special cases of the cut operation, if the first language is a right ideal or the second language is universal.

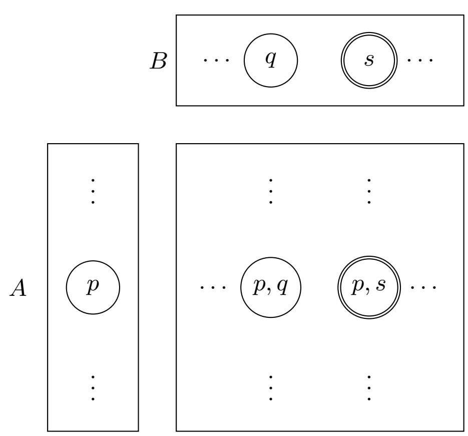

Fig. 3. The DFA B and the DFA for $L(A)\,!\,L(B)$, one row is shown. We have $p \in Q_A$ and $q, s \in Q_B$. All states in a single column have the same finality.

Lemma 1. *Let K be a right ideal language over Σ, that is, a non-empty language satisfying $K = K\Sigma^*$. Then*

$$K\,!\,L = \begin{cases} K, & \text{if } \lambda \in L; \\ \emptyset, & \text{otherwise.} \end{cases}$$

Proof. We have $K\,!\,L = \{\, uv \mid u \in K,\, v \in L, \text{ and } uv' \notin K \text{ for every } v' \in \mathrm{pref}(v)\,\}$ by definition. If $\lambda \in L$, then $K \subseteq K\,!\,L$ since for every $w \in K$, we have $w = w \cdot \lambda$ with $\lambda \in L$ and $\mathrm{pref}(\lambda) = \emptyset$. Let $u \in K$ and v be a non-empty word in L. Since K is a right ideal, $uv' \in K$ for every $v' \in \mathrm{pref}(v)$. Thus, the word uv cannot be in $K\,!\,L$, and the lemma follows. □

If the second language is universal, the cut operation acts like ordinary concatenation.

Lemma 2. *Let $K \subseteq \Sigma^*$. Then $K\,!\,\Sigma^* = K\Sigma^*$.*

Proof. We have $K\,!\,L \subseteq K\Sigma^*$ by definition. Let $w \in K\Sigma^*$. Then $w = uv$ for some $u \in K$ and $v \in \Sigma^*$. Let v' be the longest word in $\mathrm{pref}(v)$ such that $uv' \in K$. Then $w = uv'v''$ and $uv' \cdot v'' \in K\,!\,\Sigma^*$. Hence $K\Sigma^* \subseteq K\,!\,\Sigma^*$, and the stated claim follows. □

In [5, Theorem 1] the tight upper bound on state complexity of DFAs for the cut operation was obtained.

Theorem 3 ([5], Theorem 3.1). *Let A be an m-state and B an n-state* DFA. *Then $f(m,n)$ states, where*

$$f(m,n) = \begin{cases} m, & \text{if } n = 1; \\ (m-1)n + m, & \text{if } n \geq 2, \end{cases}$$

are sufficient and necessary in the worst case for any DFA *accepting $L(A)\,!\,L(B)$. The lower bound even holds for automata with binary input alphabet.*

Proof. If $n = 1$, then $L(B) = \emptyset$ or $L(B) = \Sigma^*$. We have $L(A)\,!\,\Sigma^* = L(A)\Sigma^*$ by Lemma 2. Since the language $L(A)\Sigma^*$ is accepted by an m-state DFA, the upper bound follows and it is met by the unary language $a^{m-1}a^*$. If $n \geq 2$, the proof continues as in [5, Theorem 3.1]. □

If we change to *unary* DFAs, that is, automata where the input alphabet Σ is a singleton set, that is, $\Sigma = \{a\}$ for some input symbol a, the descriptional complexity of the cut operation changes dramatically. The following tight bounds were shown in [5, Theorem 3.2].

Theorem 4 ([5], Theorem 3.2). *Let A be an m-state and B an n-state* DFA *accepting a unary language. Then $f_1(m,n)$ states, where*

$$f_1(m,n) = \begin{cases} 1, & \text{if } m = 1; \\ m, & \text{if } m \geq 2 \text{ and } n = 1; \\ 2m-1, & \text{if } m,n \geq 2 \text{ and } m \geq n; \\ m+n-2, & \text{if } m,n \geq 2 \text{ and } m < n, \end{cases}$$

are sufficient and necessary in the worst case for any DFA *accepting $L(A)\,!\,L(B)$.*

3 The Descriptional Complexity of the Cut Operation

This section is twofold. In the first subsection we investigate the magic number problem for the cut operation on unary regular languages, while in the second subsection we study the magic number problem for regular languages in general.

3.1 The Cut Operation on Unary Regular Languages

When working with unary DFAs we will use the notational convention on unary DFAs as proposed by Nicaud in [14] as a generalization of Chrobaks notation [4], in order to keep the presentation simple. It is not difficult to see that a unary DFA consists of a tail path, which starts from the initial state, followed by a cycle of one or more states. Therefore a unary DFA is totally determined by the number of states, the length of the tail path, and the set of final states. Let $A = (Q, \{a\}, \delta, q_0, F)$ be a unary DFA. We can identify the states of A with numbers from $[0, |Q| - 1]$ via $q \mapsto \min\{\, i \mid \delta(q_0, a^i) = q \,\}$. In particular the initial state q_0 is mapped to 0. Then the unary automaton A with $|Q|$ states, tail path length t, and set of accepting states F is referred to as $A = (|Q|, t, F)$. Unary minimal DFAs were characterized in [14, Lemma 1].

Theorem 5. *A unary automaton $A = (n, t, F)$ is minimal if and only if the two following conditions hold:*

1. *its loop is minimal, and*
2. *if $t \neq 0$, then states $n - 1$ and $t - 1$ do not have the same finality, that is, exactly one of them is accepting.*

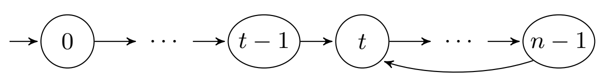

Fig. 4. Structure of a unary DFA; states $t - 1$ and $n - 1$ have opposite finality.

Now we are ready for our first result on the cut operation of unary regular languages represented by DFAs. In a series of lemmata we consider the state complexity α of the resulting automaton in increasing order of α. The first interval we are going to discuss is $[1, m]$.

Lemma 6. *Let $m, n \geq 1$. Then for every α with $1 \leq \alpha \leq m$, there exist a minimal unary m-state* DFA *A and a minimal unary n-state* DFA *B such that the minimal* DFA *for $L(A) \,!\, L(B)$ has α states.*

Proof. The proof has five cases:

1. Let $m = 1$, so we must have $\alpha = 1$. Let A be the one-state DFA accepting the empty language and B be the minimal n-state DFA for $a^{n-1}a^*$. Then $L(A) \,!\, L(B) = \emptyset$ which is accepted by a minimal one-state DFA.
2. Let $m \geq 2$, $n = 1$. By [11, Lemma 13] and Lemma 2, the result follows.
3. Let $m, n \geq 2$ and $\alpha = 1$. Consider the unary languages $a^{m-1}a^*$ and $a^{n-1}a^*$ accepted by minimal DFAs of m and n states, respectively. Then $a^{m-1}a^*$ is a right ideal and $\lambda \notin a^{n-1}a^*$. By Lemma 1, the cut of these two languages is the empty set accepted by a minimal one-state DFA.
4. Let $m \geq 2$, $n = 2$, and $2 \leq \alpha \leq m$. Consider the unary DFAs A and B defined as follows: if $m - \alpha$ is even, then $L(A) = \{\, a^i \mid i = \alpha - 2, \alpha, \ldots, m - 2 \,\}$ and $L(B) = a(aa)^*$. Otherwise, $L(A) = \{\, a^i \mid i = \alpha - 1, \alpha + 1, \ldots, m - 2 \,\}$ and set $L(B) = (aa)^*$. The minimal DFAs A and B have m and 2 states, respectively, and $L(A) \,!\, L(B) = a^{\alpha-1}(aa)^*$, which is accepted by a minimal α-state DFA—see, Figure 5.

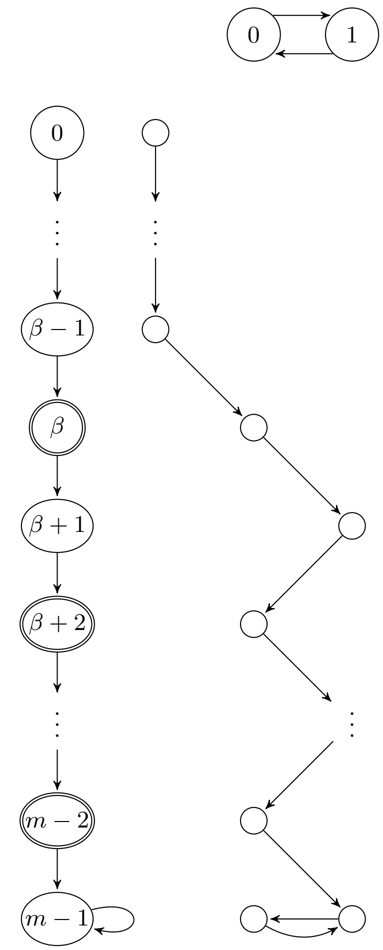

Fig. 5. The DFAs for the case 4 of Lemma 6. We have $\beta = \alpha - 2$ if $m - \alpha$ is even, and $\beta = \alpha - 1$ otherwise. The minimized DFA has α states.

5. Let $m \geq 2$, $n \geq 3$, and $2 \leq \alpha \leq m$. Consider the unary DFAs $A = (m, \alpha{-}2, [\alpha - 1, m - 1])$ and $B = (n, n{-}1, [0, n{-}2])$. Since the loop of A contains a single non-final state and the loop of B is a dead state, by Theorem 5, the DFAs A and B are minimal. The cut-construction applied to A and B results in the DFA $(m + 1, \alpha - 1, [\alpha - 1, m])$ which is equivalent to the minimal DFA $(\alpha, \alpha - 1, \{\alpha - 1\})$. □

Our next interval is $[m + 1, 2m - 1]$; compare with Theorem 4.

Lemma 7. *Let $m, n \geq 2$. Then for every α with $m + 1 \leq \alpha \leq 2m - 1$, there exist a minimal unary m-state* DFA *A and a minimal unary n-state* DFA *B such that the minimal* DFA *for $L(A)\,!\,L(B)$ has α states.*

Proof. We have $\alpha = m + \beta$ for some integer β with $1 \leq \beta \leq m - 1$. Consider the unary DFA

$$A = (m, 0, \{\beta\}).$$

Define the unary DFA B as follows:

$$B = \begin{cases} (n, 0, \{m - 1\}), & \text{if } m < n; \\ (n, n - 1, \{n - 1\}), & \text{otherwise.} \end{cases}$$

By Theorem 5, the DFAs A and B are minimal. Then we apply the cut construction to A and B and minimize the result. If $m < n$, then $L(A)\,!\,L(B)$ is accepted by the unary DFA $(\alpha, \beta, \{\alpha-1\})$, otherwise, it is accepted by the DFA $(\alpha, \beta, \{\, i \mid n+\beta-1 \leq i \leq \alpha-1 \,\})$. The resulting DFA is minimal again by Theorem 5. ◻

The last interval we are considering in this series of lemmata is $[n, m+n-2]$.

Lemma 8. *Let $m, n \geq 2$. Then for every α with $n \leq \alpha \leq m+n-2$, there exist a minimal unary m-state* DFA *A and a minimal unary n-state* DFA *B such that the minimal* DFA *for $L(A)\,!\,L(B)$ has α states.*

Proof. Consider the DFAs $A = (m, m-1, \{m-2\})$ and $B = (n, 0, \{\alpha-m+1\})$ which are minimal by Theorem 5. The cut automaton $(m+n-1, m-1, \{\alpha-1\})$ for $L(A)\,!\,L(B)$ is equivalent to the minimal DFA $(\alpha, \alpha-n, \{\alpha-1\})$. ◻

For certain values of m an n the intervals stated in the previous lemmata may not be continuous. For instance, if we choose $m = 2$ and $n = 5$, then the intervals from Theorem 10 cover $[1,3] \cup [5]$. Hence the value 4 is missing, which comes from the interval $[2m, n-1]$. In fact, we show that whenever this interval is non-empty, these values cannot be obtained by an application of the cut operation on DFAs with an appropriate number of states. Hence we have found some magic numbers.

Theorem 9 (Magic numbers in the unary case). *Let $m, n \geq 2$ satisfying $2m \leq n-1$. Then for every α with $2m \leq \alpha \leq n-1$, there exist* no *minimal unary m-state* DFA *A and minimal unary n-state* DFA *B such that the minimal* DFA *for $L(A)\,!\,L(B)$ has α states.*

Proof. We discuss two cases depending on whether the language $L(A)$ is infinite or finite. If $L(A)$ is infinite, then A must have a final state in its loop. Denote the size of loop in A by ℓ and the smallest final state in the loop of A by j. The cut construction for A and B results in a DFA $(j+\ell, j, F)$, for some set F. Since $j \leq m-1$ and $\ell \leq m$, the cut automaton has at most $2m-1$ states.

Let $A = (m, m-1, F)$ and $B = (n, k, F')$ be minimal unary DFAs such that $L(A)$ is finite. Then the state $m-1$ is a non-final sink state in A, and the state $m-2$ is final in A. It follows that in the cut automaton for A and B, the state $(m-2, 0)$ and the states $(m-1, j)$, for $j = 1, 2, \ldots, n-1$ are reachable. Two distinct states $(m-1, j)$ and $(m-1, j')$ are distinguishable by the same word as the states j and j' in B, and the state $(m-2, 0)$ and a state $(m-1, j)$ are distinguishable by the same word as 0 and j are distinguishable in B. It follows that the cut automaton has at least n reachable and pairwise distinguishable states, and the theorem follows. ◻

Now let us summarize the attainable complexities in the unary case.

Theorem 10 (Attainable complexities in the unary case). *For every $m, n, \alpha \geq 1$ such that*

1. *$\alpha = 1$, if $m = 1$,*
2. *$1 \leq \alpha \leq m$, if $m \leq 2$ and $n = 1$, or*
3. *$1 \leq \alpha \leq 2m-1$ or $n \leq \alpha \leq m+n-2$, if $m, n \geq 2$,*

there exist a minimal unary m-state DFA *A and a minimal unary n-state* DFA *B such that the minimal* DFA *for $L(A)\,!\,L(B)$ has α states.*

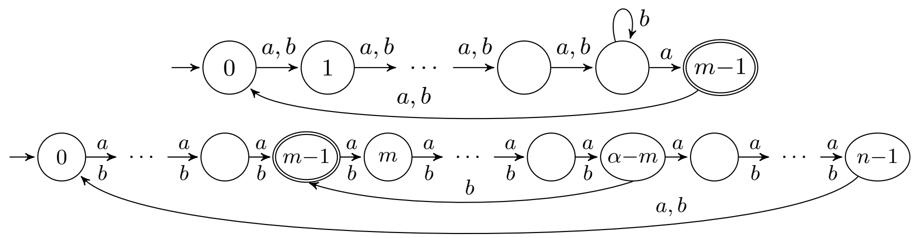

Fig. 6. The DFAs A (top) and B (bottom) for the case of $m < n$ and α satisfying $2m \le \alpha \le m + n - 1$.

Proof. The cases of $m = 1$ and $m \ge 2$, $n = 1$ are given by Lemma 6(1) and by Lemma 6(2), respectively. Let $m, n \ge 2$. The case of $1 \le \alpha \le m$ is investigated in Lemma 6(3) up to 6(5). The case of $m + 1 \le \alpha \le 2m - 1$ is given by Lemma 7. The case of $n \le \alpha \le m + n - 2$ is shown in Lemma 8. $\square$

3.2 The Cut Operation on Binary Regular Languages

Next we consider the magic number problem for regular languages in general. Here it will turn out that no magic numbers exist in this case, even for languages on a binary alphabet. First, we show that the values from $[1, m + n - 1]$ are not magic in the binary case.

Lemma 11. *Let $m, n \ge 2$ and $m < n$. Then for α with $2m \le \alpha \le m + n - 1$, there exist a minimal binary m-state* DFA A *and a minimal binary n-state* DFA B *such that the minimal* DFA *for $L(A)\,!\,L(B)$ has α states.*

Proof. Consider the binary DFA $A = ([0, m-1], \{a, b\}, \delta_A, 0, \{m-1\})$, where

$$\delta_A(i, a) = (i + 1) \bmod m \quad \text{and} \quad \delta_A(i, b) = \begin{cases} (i+1) \bmod m, & \text{if } i \ne m-2, \\ m-2, & \text{otherwise.} \end{cases}$$

Next, consider the binary DFA $B = ([0, n-1], \{a, b\}, \delta_B, 0, \{m-1\})$, where

$$\delta_B(j, a) = (j + 1) \bmod n \quad \text{and} \quad \delta_B(j, b) = \begin{cases} (j+1) \bmod n, & \text{if } j \ne \alpha - m, \\ m-1, & \text{otherwise.} \end{cases}$$

Both automata are depicted in Figure 6 and the cut automaton is drawn in Figure 7. In the cut automaton for $L(A)\,!\,L(B)$ we consider the following set of states:

$$\begin{aligned} \mathcal{R}_1 &= \{\, (i, \bot) \mid 0 \le i \le m-2 \,\} \cup \{(m-1, 0)\} \cup \{\, (i, i+1) \mid 0 \le i \le m-3 \,\}, \\ \mathcal{R}_2 &= \{(m-2, m-1)\}, \\ \mathcal{R}_3 &= \{\, (m-2, j) \mid m \le j \le \alpha - m \,\}, \end{aligned}$$

and let $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2 \cup \mathcal{R}_3$. Each state in $\mathcal{R}_1 \cup \mathcal{R}_2$ is reached from $(0, \bot)$ by a word in a^*, and each state in $\mathcal{R}_3$ is reached from $(m-2, m-1)$ by a word in b^*. Next, notice that each state in $\mathcal{R}$ goes to a state in $\mathcal{R}$ on both a and b. It follows that no other state is reachable in the cut automaton. To prove distinguishability, notice that the state $(m-2, m-1)$ is a unique final state. Next, two distinct states in $\mathcal{R}_1$ are distinguishable by a word in a^* and two distinct states in $\mathcal{R}_3$ are distinguishable by a word in b^*. The state $(m-2, \bot)$ is distinguishable from a state in $\mathcal{R}_3$ by a word in b^*. Every other state in $\mathcal{R}_1$ is distinguishable from a state in $\mathcal{R}_3$ since the word a leads them to two distinct states in $\mathcal{R}_1$ which are distinguishable as shown above. Since $|\mathcal{R}| = \alpha$, our proof is complete. $\square$

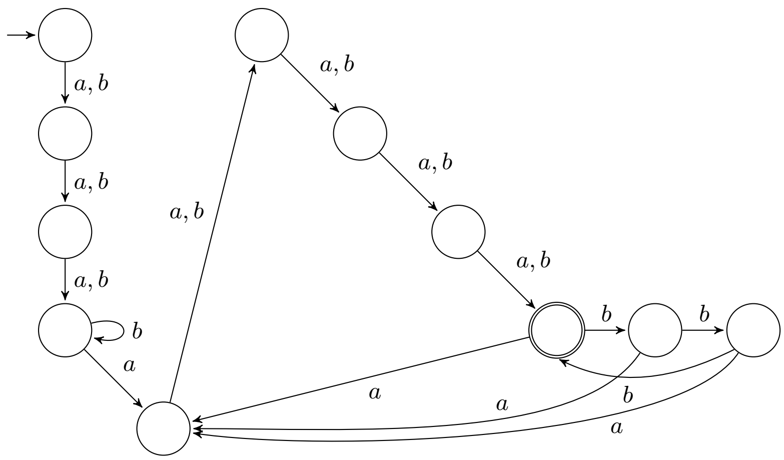

Fig. 7. The cut automaton for the DFAs from Figure 6 if $m = 5$, $n = 9$, and $\alpha = 11$.

Since the operational complexity of the cut operation for regular languages in general is higher than those for unary language, we have to consider the remaining interval $[m+n-1, (m-1)n+m]$ up to the maximal state complexity. This is done in three steps—compare with [8]:

1. First we show that some special values of α are attainable, namely

$$\alpha = 1 + (m-1)s + (n-s)r$$

   for some r, s with $1 \leq r \leq m-1$ and $1 \leq s \leq n$; every such value corresponds to the number of states in the cut automaton in which the following states are reachable: the state $(0,0)$, all the states in rows $1, 2, \ldots, r$, and all the states in columns $0, 1, \ldots, s-1$ except for those in row 0. Here the term rows and columns refers to the state components.
2. Then we show that all the remaining values of α in $[m+n-1, (m-1)n+1]$ are attainable.
3. Finally, we show that all the values of α in $[(m-1)n+2, (m-1)n+m]$ are attainable.

Let us start with the first task.

Lemma 12. *Let $m, n \geq 2$ and set r, s such that $1 \leq r \leq m-1$ and $1 \leq s \leq n$. Then there exist a minimal binary m-state* DFA *$A_{r,s}$ and a minimal binary n-state* DFA *$B_{r,s}$ such that the minimal* DFA *for $L(A_{r,s})\,!\,L(B_{r,s})$ has exactly $1 + (m-1)s + (n-s)r$ states.*

Proof. Our aim is to define the DFAs

$$A_{r,s} = ([0, m-1], \{a, b\}, \delta_A, 0, \{0\})$$

and

$$B_{r,s} = ([0, n-1], \{a, b\}, \delta_B, 0, \{n-1\})$$

in such a way that in the DFA for the cut of their languages the states in the following sets would be reachable and pairwise distinguishable:

$$\begin{aligned}
\mathcal{R}_1 &= \{(0,0)\}, \\
\mathcal{R}_2 &= \{\, (i,j) \mid 1 \leq i \leq m-1 \text{ and } 0 \leq j \leq s-1 \,\}, \\
\mathcal{R}_3 &= \{\, (i,j) \mid 1 \leq i \leq r \text{ and } s \leq j \leq n-1 \},
\end{aligned}$$

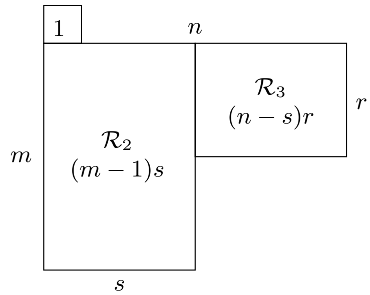

Fig. 8. A part of a cut automaton with $\alpha_{r,s} = 1 + (m-1)s + (n-s)r$ states.

and let $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2 \cup \mathcal{R}_3$. For a schematic illustration of these sets within the cut automaton see Figure 8. Figure 9 shows the binary DFAs from Lemma 12 for $m+n-1$ up to the value $(m-1)n+1$ and the resulting cut automaton—compare with Figure 10 from Lemma 13. Moreover, we have to assure that no other state of the cut automaton is reachable. Since

$$|\mathcal{R}_1| + |\mathcal{R}_2| + |\mathcal{R}_3| = 1 + (m-1)s + (n-s)r,$$

the DFAs $A_{r,s}$ and $B_{r,s}$ will be the desired DFAs. To this aim, we define δ_A and δ_B as follows:

$$\delta_A(i,a) = (i+1) \bmod m \quad \text{and} \quad \delta_A(i,b) = \begin{cases} i & \text{if } i \le r; \\ r & \text{if } i \ge r+1; \end{cases}$$

and

$$\delta_B(j,b) = (j+1) \bmod n \quad \text{and} \quad \delta_B(j,a) = \begin{cases} j & \text{if } j \le s-1; \\ s-1 & \text{if } j \ge s. \end{cases}$$

Then, in the DFA for $L(A_{r,s}) \,!\, L(B_{r,s})$, the state $(0,0)$ is the initial state, and each state (i,j) in $\mathcal{R}$ is reached from $(0,0)$ by $a^i b^j$. To show that no other state is reachable, notice that each state (i,j) in $\mathcal{R}$ goes on a to a state (i',j') where $j' \le s-1$, and it goes on b to a state (i'',j'') where $i'' \le r$. Since both resulting states are in $\mathcal{R}$, no other state is reachable in the cut automaton.

It remains to prove the distinguishability of states in $\mathcal{R}$. The state $(0,0)$ and any other state in $\mathcal{R}$ are distinguishable by a word in b^*. Two states in different columns are distinguishable by a word in b^* since exactly one of them can be moved to the last column containing the final states of the cut automaton. Two states in different rows are distinguishable by a word in a^* since exactly one of them can be moved to the state $(0,0)$. This proves distinguishability and concludes the proof. □

In the above lemma we obtained the values

$$\alpha_{r,s} = 1 + (m-1)s + (n-s)r$$

in $[m+n-1, (m-1)n+1]$. We still need to get the values between $\alpha_{r,s}$ and $\alpha_{r+1,s}$ resp. $\alpha_{r,s+1}$ We have $\alpha_{r+1,s} - \alpha_{r,s} = n-s$ and $\alpha_{r,s+1} - \alpha_{r,s} = m-1-r$, so we need to obtain the complexities $\alpha_{r,s} + t$, where $1 \le t \le \min\{n-s, m-1-r\} - 1$. The next lemma produces these complexities.

Lemma 13. *Let $m,n \ge 2$ and set r,s such that $1 \le r \le m-1$ and $1 \le s \le n$. Moreover let t satisfy $1 \le t \le \min\{n-s,\ m-1-r\} - 1$. Then there exist a minimal binary m-state* DFA $A_{r,s,t}$

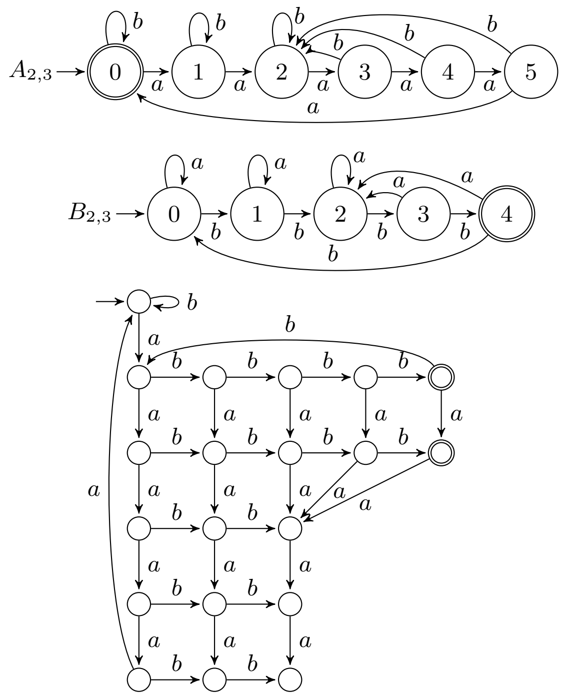

Fig. 9. The DFAs $A_{2,3}$ and $B_{2,3}$; $m = 6$, $n = 5$ (left). The DFA for $L(A_{2,3})\,!\,L(B_{2,3})$ (right). Not all transitions are shown.

and a minimal binary n-state DFA $B_{r,s,t}$ *such that the minimal* DFA *for* $L(A_{r,s,t})\,!\,L(B_{r,s,t})$ *has* $1 + (m-1)s + (n-s)r + t$ *states.*

Proof. Let $\alpha_{r,s} = 1 + (m-1)s + (n-s)r$. Then in the cut automaton for the languages accepted by DFAs $A_{r,s}$ and $B_{r,s}$ described in the previous proof, exactly $\alpha_{r,s}$ states are reachable and distinguishable. Our aim is to modify both automata in such a way that in the resulting cut automaton has t more reachable states $q_1, q_2, \ldots, q_t$. We want these states to be in the row $r+1$ or in the column s. Moreover, we have to be careful and assure that no other state is reachable. Since we have only two letters, the states q_i must not be neighboring. For that reason, the states $q_1, q_3, \ldots$ are in the row $r+1$ and columns $s+1, s+3, \ldots$, and the states $q_2, q_4, \ldots$ are in the column s and rows $r+2, r+4, \ldots$.

To achieve this goal, we modify DFAs $A_{r,s}$ and $B_{r,s}$ as follows: in $A_{r,s}$, we replace each transition $(r+i-1, b, r)$ by $(r+i-1, b, r+i)$, if $1 \le i \le t$ and i is even. In $B_{r,s}$, we replace each transition $(s+i-1, a, s-1)$ by $(s+i-1, a, s+i)$, if $1 \le i \le t$ and i is odd. Denote the resulting DFAs by $A_{r,s,t}$ and $B_{r,s,t}$, respectively; see Figure 10 for an illustration in case $m = 7$, $n = 8$, $r = 2$, $s = 3$, and $t = 3$.

Consider the cut automaton for $L(A_{r,s,t})\,!\,L(B_{r,s,t})$. Let $\mathcal{R}$, $\mathcal{R}_1$, $\mathcal{R}_2$, $\mathcal{R}_3$ be the same sets as in the previous proof. Then each state (i,j) in $\mathcal{R}$ is reachable from $(0,0)$ by $a^i b^j$. Next, if i is odd, then each state $q_i = (r+1, s+i)$ is reached from $(r, s+i-1)$ by a, and if i is even, then each state $q_i = (r+i, s)$ is reached from $(r+i-1, s-1)$ by b. Now, let us show that that no other state is reachable. Notice that each state in $\mathcal{R}$ goes either to a state in $\mathcal{R}$ or to a state in $\{q_1, q_2, \ldots, q_t\}$ on a and b. Next, each state in $\{q_1, q_2, \ldots, q_t\}$ goes to a state in row $r+1$ or to a state in column s on a and b. Since all the resulting states are in $\mathcal{R} \cup \{q_1, q_2, \ldots, q_t\}$, no other state is reachable in the cut automaton. The proof of distinguishability is exactly the same as in the proof of Lemma 12. $\square$

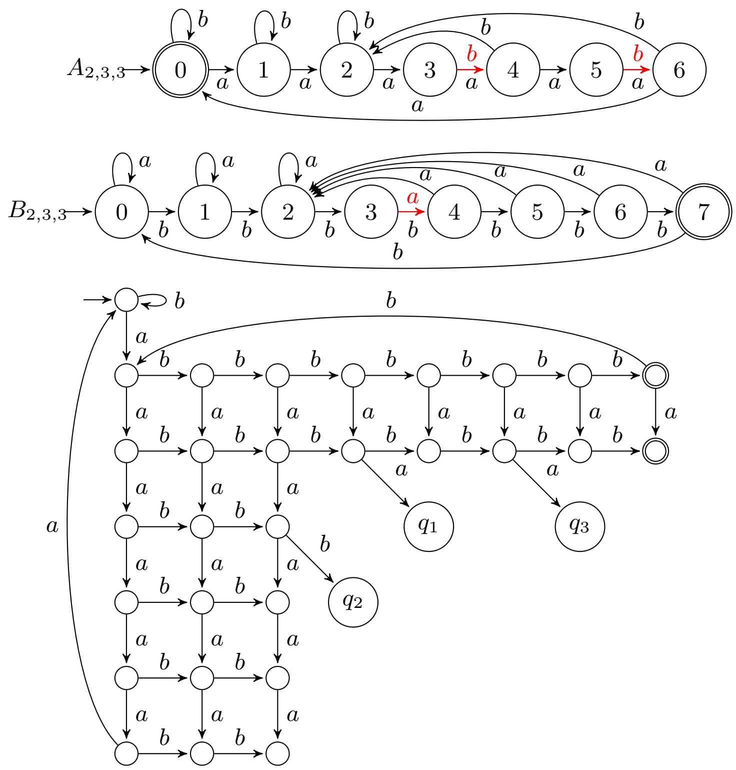

Fig. 10. For $m = 7$ and $n = 8$, the DFAs $A_{2,3,3}$ and $B_{2,3,3}$ are shown on top and the cut automaton for $L(A_{2,3,3})\,!\,L(B_{2,3,3})$ is depicted below. Not all transitions are shown.

We have already produced all the complexities in $[m+n-1, (m-1)n+1]$. It remains to show that the complexities in $[(m-1)n+2, (m-1)n+m]$ are attainable.

Lemma 14. *Let $m, n \geq 2$ and $(m-1)n+2 \leq \alpha \leq (m-1)n+m$. There exist a minimal binary m-state* DFA *A and a minimal binary n-state* DFA *B such that the minimal* DFA *for $L(A)\,!\,L(B)$ has α states.*

Proof. We have $\alpha = (m-1)n + 1 + \beta$ for some β such that $1 \leq \beta \leq m-1$. Let $B = B_{m-1,n}$, where DFA $B_{m-1,n}$ is given by Lemma 12. Let A be the DFA obtained from $A_{m-1,n}$ by making the state 0 non-final and by making the state β final; see Figure 11 for an illustration. Consider the cut automaton constructed from A and B. Denote

$$\mathcal{R}_1 = \{\, (i, \perp) \mid i \in [0, \beta-1] \,\} \cup \{(\beta, 0)\},$$
$$\mathcal{R}_2 = \{\, (i, j) \mid i \in [0, \beta-1] \cup [\beta+1, m-1] \text{ and } j \in [0, n-1] \,\}.$$

Notice that each state in $\mathcal{R}_1$ is reachable since by reading β consecutive a's we start in $(0, \perp)$ and run through the states $(1, \perp), \ldots, (\beta-1, \perp)$ followed by $(\beta, 0)$, and ending in $(0, 0)$ by further reading $a^{m-\beta}$. Each state (i, j) in $\mathcal{R}_2$ is reached from $(0, 0)$ by $a^i b^j$. No other state may be reachable in the cut automaton.

To prove distinguishability, let p and q be two different states in $\mathcal{R}_1 \cup \mathcal{R}_2$. If $p \in \mathcal{R}_1$ and $q \in \mathcal{R}_2$, then p is a non-final state with a loop on b, while a word in b^* is accepted from q. If both p and q are in $\mathcal{R}_1$, then a word in a^* leads one of them to the state $((\beta+1) \bmod m, 0)$ in $\mathcal{R}_2$, while it leads the second one to a state in $\mathcal{R}_1$, and the resulting states are distinguishable as shown above. Finally, let p and q be in $\mathcal{R}_2$. If they are in different columns, then a word in b^*

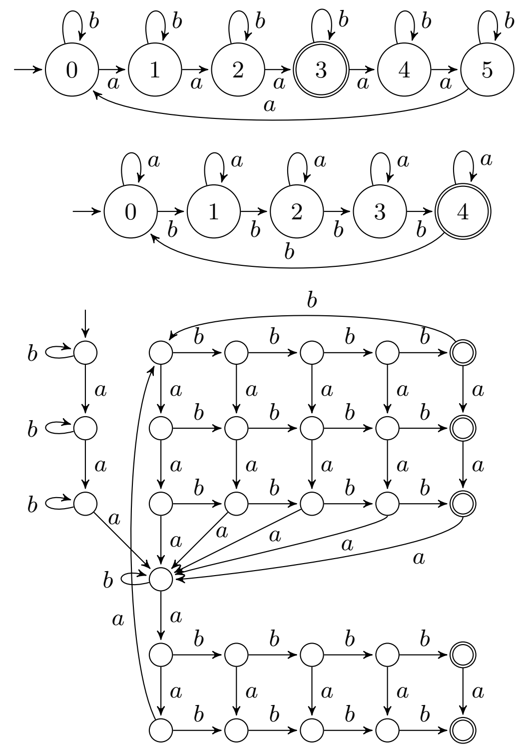

Fig. 11. The DFAs A and B; $m = 6$, $n = 5$, and $\beta = 3$ (left). The DFA for $L(A)\,!\,L(B)$ (right). Not all transitions are shown.

leads one of them to column $n - 1$ in which all states are final, while it leads the second one to a different column in which all states are non-final. If p and q are in different rows, then a word in a^* leads one of them to the state $(\beta, 0)$ in $\mathcal{R}_1$, and it leads the second one to a state in $\mathcal{R}_2$. $\square$

We summarize the result of this section.

Theorem 15 (Attainable complexities in the general case). *Let $m, n \geq 1$ and $f(m, n)$ be the state complexity of the cut operation. For each α such that $1 \leq \alpha \leq f(m, n)$, there exist a minimal binary m-state* DFA *A and a minimal binary n-state* DFA *B such that the minimal* DFA *for $L(A)\,!\,L(B)$ has α states.*

Proof. First notice that if we have unary witnesses over $\{a\}$ for some complexity of their cut, and define transitions on b identically as on a, we get binary witnesses for the same complexity. The case of $n = 1$ follows from Lemma 6(1) and 6(2). Let $n \geq 2$. The case of $m = 1$ is given by Lemma 6(1) since $f(1, n) = 1$. Let $m \geq 2$. The case of $1 \leq \alpha \leq m$ is given by Lemma 6(3)–6(5), and the case of $m + 1 \leq \alpha \leq 2m - 1$ by Lemma 7. The case of $2m \leq \alpha \leq m + n - 1$ is given by Lemma 11. The complexities from $[m + n - 1, (m - 1)n + 1]$ which can be written in the form $1 + (m - 1) \cdot s + (n - s) \cdot r$ are covered by Lemma 12, and all the remaining complexities in this set are covered by Lemma 13. Finally, the complexities in $[(m - 1)n + 1, (m - 1)n + m]$ are covered by Lemma 14. $\square$

Observe, that the previous theorem solves the magic number problem for the cut operation for arbitrary sized alphabets of at least two letters, since adding duplicitous letters does not change the complexity.

Acknowledgements. We would like to thank Juraj Šebej and Jozef Jirásek Jr. for their help with proving the validity of our theorems for border values. Moreover, also thanks to Galina Jirásková for her support on the paper. We also thank the anonymous referee who pointed out a flaw in an earlier version of the proof of Lemma 6.

References

1. Berglund, M., Björklund, H., Drewes, F., van der Merwe, B., Watson, B.: Cuts in regular expressions. In: 17th DLT. pp. 70–81. No. 7907 in LNCS, Springer, Marne-la-Vallée, France (2011)
2. Čevorová, K., Jirásková, G., Krajňáková, I.: On the square of regular languages. In: 19th CIAA. pp. 136–147. No. 8587 in LNCS, Springer, Giessen, Germany (2014)
3. Čevorová, K.: Kleene star on unary regular languages. In: 15th DFCS. pp. 277–288. No. 8031 in LNCS, Springer, London, Ontario, Canada (2013)
4. Chrobak, M.: Finite automata and unary languages. Theoret. Comput. Sci. 47, 149–158 (1986)
5. Drewes, F., Holzer, M., Jakobi, S., van der Merwe, B.: Tight bounds for cut-operations on deterministic finite automata. Fundam. Inform. 153, 1–22 (2017)
6. Geffert, V.: Magic numbers in the state hierarchy of finite automata. Inform. Comput. 205(11), 1652–1670 (2007)
7. Harrison, M.A.: Introduction to Formal Language Theory. Addison-Wesley (1978)
8. Hricko, M., Jirásková, G., Szabari, A.: Union and intersection of regular languages and descriptional complexity. In: 7th DCFS. pp. 170–181. Universita degli Studi di Milano, Milano, Italy (2005)
9. Iwama, K., Kambayashi, Y., Takaki, K.: Tight bounds on the number of states of DFAs that are equivalent to n-state NFAs. Theoret. Comput. Sci. 237(1–2), 485–494 (2000)
10. Jirásková, G.: Magic numbers and ternary alphabet. Internat. J. Found. Comput. Sci. 22(2), 331–344 (2011)
11. Jirásková, G., Szabari, A., Šebej, J.: The complexity of languages resulting from the concatenation operation. In: 18th DFCS. pp. 153–167. No. 9777 in LNCS, Springer, Bucharest, Romania (2016)
12. Meyer, A.R., Fischer, M.J.: Economy of description by automata, grammars, and formal systems. In: 12th SWAT. pp. 188–191. IEEE Computer Society Press (1971)
13. Moore, F.R.: On the bounds for state-set size in the proofs of equivalence between deterministic, nondeterministic, and two-way finite automata. IEEE Transaction on Computing C-20, 1211–1219 (1971)
14. Nicaud, C.: Average state complexity of operations on unary automata. In: 24th MFCS. pp. 231–240. No. 1672 in LNCS, Springer, Szklarska Poreba, Poland (1999)
15. Rabin, M.O., Scott, D.: Finite automata and their decision problems. IBM J. Res. Dev. 3, 114–125 (1959)

Institut für Informatik

Justus-Liebig-Universität Giessen
Arndtstr. 2, 35392 Giessen, Germany

Recent Reports

(Further reports are available at `www.informatik.uni-giessen.de`.)

M. Holzer, S. Jakobi, *A Note on the Computational Complexity of Some Problems for Self-Verifying Finite Automata*, Report 1702, April 2017.

S. Beier, M. Holzer, M. Kutrib, *On the Descriptional Complexity of Operations on Semilinear Sets*, Report 1701, April 2017.

M. Holzer, S. Jakobi, M. Wendlandt, *On the Computational Complexity of Partial Word Automata Problems*, Report 1404, May 2014.

H. Gruber, M. Holzer, *Regular Expressions From Deterministic Finite Automata, Revisited*, Report 1403, May 2014.

M. Kutrib, A. Malcher, M. Wendlandt, *Deterministic Set Automata*, Report 1402, April 2014.

M. Holzer, S. Jakobi, *Minimal and Hyper-Minimal Biautomata*, Report 1401, March 2014.

J. Kari, M. Kutrib, A. Malcher (Eds.), *19th International Workshop on Cellular Automata and Discrete Complex Systems AUTOMATA 2013 Exploratory Papers*, Report 1302, September 2013.

M. Holzer, S. Jakobi, *Minimization, Characterizations, and Nondeterminism for Biautomata*, Report 1301, April 2013.

A. Malcher, K. Meckel, C. Mereghetti, B. Palano, *Descriptional Complexity of Pushdown Store Languages*, Report 1203, May 2012.

M. Holzer, S. Jakobi, *On the Complexity of Rolling Block and Alice Mazes*, Report 1202, March 2012.

M. Holzer, S. Jakobi, *Grid Graphs with Diagonal Edges and the Complexity of Xmas Mazes*, Report 1201, January 2012.

H. Gruber, S. Gulan, *Simplifying Regular Expressions: A Quantitative Perspective*, Report 0904, August 2009.

M. Kutrib, A. Malcher, *Cellular Automata with Sparse Communication*, Report 0903, May 2009.

M. Holzer, A. Maletti, *An $n \log n$ Algorithm for Hyper-Minimizing States in a (Minimized) Deterministic Automaton*, Report 0902, April 2009.

H. Gruber, M. Holzer, *Tight Bounds on the Descriptional Complexity of Regular Expressions*, Report 0901, February 2009.

M. Holzer, M. Kutrib, and A. Malcher (Eds.), *18. Theorietag Automaten und Formale Sprachen*, Report 0801, September 2008.

M. Holzer, M. Kutrib, *Flip-Pushdown Automata: Nondeterminism is Better than Determinism*, Report 0301, February 2003

M. Holzer, M. Kutrib, *Flip-Pushdown Automata: $k + 1$ Pushdown Reversals are Better Than k*, Report 0206, November 2002

M. Holzer, M. Kutrib, *Nondeterministic Descriptional Complexity of Regular Languages*, Report 0205, September 2002

H. Bordihn, M. Holzer, M. Kutrib, *Economy of Description for Basic Constructions on Rational Transductions*, Report 0204, July 2002

M. Kutrib, J.-T. Löwe, *String Transformation for n-dimensional Image Compression*, Report 0203, May 2002

A. Klein, M. Kutrib, *Grammars with Scattered Nonterminals*, Report 0202, February 2002

A. Klein, M. Kutrib, *Self-Assembling Finite Automata*, Report 0201, January 2002

M. Holzer, M. Kutrib, *Unary Language Operations and its Nondeterministic State Complexity*, Report 0107, November 2001

A. Klein, M. Kutrib, *Fast One-Way Cellular Automata*, Report 0106, September 2001

M. Holzer, M. Kutrib, *Improving Raster Image Run-Length Encoding Using Data Order*, Report 0105, July 2001

M. Kutrib, *Refining Nondeterminism Below Linear-Time*, Report 0104, June 2001

M. Holzer, M. Kutrib, *State Complexity of Basic Operations on Nondeterministic Finite Automata*, Report 0103, April 2001

M. Kutrib, J.-T. Löwe, *Massively Parallel Fault Tolerant Computations on Syntactical Patterns*, Report 0102, March 2001