Incomplete Operational Transition Complexity of Regular Languages

Eva Maia[a], Nelma Moreira[a], Rogério Reis[a]

[a] *CMUP & DCC, Faculdade de Ciências da Universidade do Porto*
Rua do Campo Alegre, 4169-007 Porto, Portugal

Abstract

The state complexity of basic operations on regular languages considering complete deterministic finite automata (DFA) has been extensively studied in the literature. But, if incomplete DFAs are considered, transition complexity is also an significant measure. In this paper we study the incomplete (deterministic) state and transition complexity of some operations for regular and finite languages. For regular languages we give a new tight upper bound for the transition complexity of the union, which refutes the conjecture presented by Y. Gao et al.. For finite languages, we correct the published state complexity of concatenation for complete DFAs and provide a tight upper bound for the case when the *right* operand is larger than the *left* one. We also present some experimental results to test the behaviour of those operations on the average case, and we conjecture that for many operations and in practical applications the worst-case complexity is seldom reached.

Keywords: Descriptional complexity, Automata Theory, Regular Languages, Transition Complexity, State Complexity
2010 MSC: 68Q45

1. Introduction

In the last two decades the descriptional complexity of regular languages has been extensively investigated. For deterministic finite automata (DFA), the complexity measure usually studied is the state complexity, i.e. the number of states of the complete minimal DFA, [3, 23, 24, 11, 4, 25], while for nondeterministic finite automata (NFA) both state and transition complexity were considered [10, 7, 19, 12, 11]. For NFAs transition complexity is generally

☆This work was partially funded by the European Regional Development Fund through the programme COMPETE and by the Portuguese Government through the FCT under project PEst-C/MAT/UI0144/2013 and project FCOMP-01-0124-FEDER-020486. Eva Maia is funded by FCT grant SFRH/BD/78392/2011.

Email addresses: `emaia@dcc.fc.up.pt` (Eva Maia), `nam@dcc.fc.up.pt` (Nelma Moreira), `rvr@dcc.fc.up.pt` (Rogério Reis)

considered a more interesting measure. Considering complete DFAs, where the transition function is total, the transition complexity is, obviously, the product of the alphabet size by the state complexity. But in many applications where large alphabets need to be considered or, in general, when very sparse transition functions take place, partial transition functions are very convenient. Examples include lexical analysers, discrete event systems, or any application that uses dictionaries where compact automaton representations are essential, for instance for manipulation on large Unicode alphabets [2, 17, 6, 18]. And, in many cases, only finite languages are needed. Thus, it makes sense to investigate the transition complexity of not necessarily complete DFAs.

In this paper we study the incomplete operational transition complexity of several operations on regular and finite languages. To be comprehensive we also analyse the state complexity of resulting languages. In general, transition complexity bounds depend not only on the complexities of the operands but also on other refined measures, as the number of undefined transitions or the number of transitions that leave the initial state. For both families of languages we performed some experimental tests in order to have an idea of the average-case complexity of those operations.

The paper is organized as follows. Section 2 recalls some useful definitions and notation. In Section 3, we study the state and transition complexity for the union, concatenation, Kleene star and reversal operations on regular languages. For all these operations tight upper bounds are given. The tight upper bound presented for the transition complexity of the union operation refutes the conjecture presented by Y.Gao *et al.* [8]. We also present the same study for unary regular languages. In Subsection 3.6 we analyse some experimental results. In the Section 4 we continue the line of research of the Section 3 considering finite languages. For the concatenation, we correct the upper bound for the state complexity of complete DFAs [5], and show that if the *right* operand is larger than the *left* one, the upper bound is only reached using an alphabet of variable size. We also present some experimental results for finite languages. The algorithms and the witness language families used in this work, although new, are based on the ones of Yu et al. [26]; several proofs required new techniques.

Table 1 presents a summary and a comparison of the obtained results for transition complexity on general and finite languages. Note that the values in the table are obtained using languages for which the upper bounds are reached. This paper expands the work presented in extended abstracts [16, 15] with full proofs of theorems and experimental tests.

2. Preliminaries

We recall some basic notions about finite automata and regular languages. For more details, we refer the reader to the standard literature [13, 22, 21].

Given two integers $m, n \in \mathbb{N}$, let $[m, n] = \{i \in \mathbb{N} \mid m \leq i \leq n\}$ and $[m, n[= \{i \in \mathbb{N} \mid n \leq i < n\}$.

A DFA is a five-tuple $A = (Q, \Sigma, \delta, q_0, F)$ where Q is a finite set of states, Σ is a finite input alphabet, δ is the transition function $\delta : Q \times \Sigma \to Q$, q_0

Operation	Regular	$\|\Sigma\|$	Finite	$\|\Sigma\|$
$L_1 \cup L_2$	$2n(m+1)$	2	$3(mn-n-m)+2$	$f_1(m,n)$
$L_1 \cap L_2$	nm	1	$(m-2)(n-2)(2+\sum_{i=1}^{\min(m,n)-3}(m-2-i)(n-2-i))+2$	$f_2(m,n)$
L^C	$m+2$	1	$m+1$	1
L_1L_2	$2^{n-1}(6m+3)-5$, if $m,n \geq 2$	3	$2^n(m-n+3)-8$, if $m+1 \geq n$	2
			See Theorem 16(7)	$n-1$
$L^\star$	$3.2^{m-1}-2$, if $m \geq 2$	2	$9 \cdot 2^{m-3} - 2^{m/2} - 2$, if m is odd	3
			$9 \cdot 2^{m-3} - 2^{(m-2)/2} - 2$, if m is even	
L^R	$2(2^m-1)$	2	$2^{p+2}-7$, if $m=2p$	2
			$3 \cdot 2^p - 8$, if $m = 2p-1$	

Table 1: Incomplete transition complexity for regular and finite languages, where m and n are the (incomplete) state complexities of the operands, $f_1(m,n) = (m-1)(n-1)+1$ and $f_2(m,n) = (m-2)(n-2)+1$. The column $|\Sigma|$ indicates the minimal alphabet size for which the upper bound is reached.

in Q is the initial state, and $F \subseteq Q$ is the set of final states. Let $|\Sigma| = k$, $|Q| = n$, and without loss of generality, we assume $Q = [0, n[$ with $q_0 = 0$. The transition function can be naturally extended to subsets of Q and words $w \in \Sigma^\star$. A DFA is called *complete* if the transition function δ is total. In this paper we consider the DFAs to be not necessarily complete, *i.e.* with partial transition functions. For $q \in Q$ and $\sigma \in \Sigma$, if $\delta(q,\sigma)$ is defined we write $\delta(q,\sigma)\downarrow$, and $\delta(q,\sigma)\uparrow$, otherwise, and, when defining a DFA, an assignment $\delta(q,\sigma) = \uparrow$ means that the transition is undefined. The *language* accepted by A is $\mathcal{L}(A) = \{w \in \Sigma^\star \mid \delta(q_0, w) \in F\}$. Two DFAs are *equivalent* if they accept the same language. For each regular language there exists a unique minimal complete DFA with the minimum number of states. The *left quotient* of $L \subseteq \Sigma^\star$ by $x \in \Sigma^\star$ is $D_xL = \{z \mid xz \in L\}$. The equivalence relation $R_L \subseteq \Sigma^\star \times \Sigma^\star$ is defined by $(x,y) \in R_L$ if and only if $D_xL = D_yL$. The *Myhill-Nerode Theorem* states that a language L is regular if and only if R_L has a finite number of equivalence classes, *i.e.*, L has a finite number of left quotients. This number is the number of states of the minimal complete DFA, whih is unique up to isomorphism. Using Myhill-Nerode theorem, it is easy to prove that an automaton is minimal if all its states correspond to different left quotients. Thus, to prove that a DFA is minimal it is enough to show that for each state q, there is a word w such that $\delta(q,w) \in F$ and $\delta(q',w) \notin F$ for all other states $q' \neq q$. We say that this word w distinguishes q from the other states. The *state complexity*, $\mathrm{sc}(L)$, of a regular language L is the number of states of the minimal complete DFA of L. If we consider the non-complete minimal DFA, its number of states is the number of left quotients minus one, due to the removal of the *dead state*, that we denote by Ω. The left quotient corresponding to Ω is the empty language. The *incomplete state complexity* of a regular language L ($\mathrm{isc}(L)$) is the number of states of the minimal DFA without states non conducting to a final state (thus, not necessarily complete) that accepts L. Note that $\mathrm{isc}(L)$ differs at most by 1 from $\mathrm{sc}(L)$ ($\mathrm{isc}(L) \in \{\mathrm{sc}(L) - 1, \mathrm{sc}(L)\}$). The *incomplete transition complexity*, $\mathrm{itc}(L)$, of a regular language L is the minimal number of transitions over all DFAs that accept L. Whenever the model is explicitly given we refer only to *state* or *transition* complexity, by omitting the term incomplete[1]. It is well known that the minimal DFA of a language has also the minimal number of transitions.

A transition labeled by $\sigma \in \Sigma$ is called a *σ-transition* (represented by $\delta(q, \sigma)$, where $q \in Q$) and the number of σ-transitions of a DFA A is denoted by $t_\sigma(A)$. The *σ-transition complexity* of L, $\mathrm{itc}_\sigma(L)$, is the minimal number of σ-transitions of any DFA recognizing L. In [8, Lemma 2.1] it was proved that the minimal DFA accepting L has the minimal number of σ-transitions, for every $\sigma \in \Sigma$. From this it follows that $\mathrm{itc}(L) = \sum_{\sigma\in\Sigma} \mathrm{itc}_\sigma(L)$.

The *complexity of an operation* on regular languages is the (worst-case) complexity of a language resulting from the operation, considered as a function of the complexities of the operands. Usually an *upper bound* is obtained by providing an algorithm which, given representations of the operands (*e.g.* DFAs), constructs a model (*e.g.* DFA) that accepts the language resulting from the referred operation. The number of states or transitions of the resulting DFA is an upper bound for the state or the transition complexity of the operation, respectively. To prove that an upper bound is *tight*, for each operand we can give a family of languages (parametrized by the complexity measures), called *witnesses*, such that the complexity of the resulting language achieves that upper bound. To express the transition complexity of a language operation, we also use the following measures and refined numbers of transitions. Let $A = (Q, \Sigma, \delta, 0, F)$ be a DFA, with $Q = [0, n[$, $\sigma \in \Sigma$, and $i \in Q$, we define

- $f(A) = |F|$;
- $t_\sigma(A, i) = \begin{cases} 1, & \text{if there exists a } \sigma\text{-transition leaving } i; \\ 0, & \text{otherwise;} \end{cases}$
- $\bar{t}_\sigma(A, i)$ is the negation of $t_\sigma(A, i)$;
- $s_\sigma(A) = t_\sigma(A, 0)$;
- $t_\sigma(A) = \sum_{i\in Q} t_\sigma(A, i)$;
- $u_\sigma(A) = |Q| - t_\sigma(A)$; and
- $\tilde{u}_\sigma(A)$ is the number of non-final states without σ-transitions.

Whenever there is no ambiguity we omit A from the above definitions. If $t_\sigma(A) = |Q|$ we say that A is *σ-complete*, and *σ-incomplete*, otherwise. All the above

[1] In [8] the authors use the notation $\mathrm{sc}(L)$ and $\mathrm{tc}(L)$ instead of $\mathrm{isc}(L)$ and $\mathrm{itc}(L)$.

Operation	sc	isc	nsc
$L_1 \cup L_2$	mn	$\mathbf{mn + m + n}$	$m + n + 1$
$L_1 \cap L_2$	mn	mn	mn
L^C	n	$n + 1$	2^n
$L_1 L_2$	$m2^n - f_1 2^{n-1}$	$\mathbf{(m+1)2^n - f_1 2^{n-1} - 1}$	$m + n$
$L^\star$	$2^{m-1} + 2^{m-l-1}$	$\mathbf{2^{m-1} + 2^{m-l-1}}$	$m + 1$
L^R	2^m	$\mathbf{2^m - 1}$	$m + 1$

Table 2: State complexity of basic regularity preserving operations on regular languages.

Operation	itc	ntc
$L_1 \cup L_2$	$\mathrm{itc}(\mathbf{L_1})(\mathbf{1 + n}) + \mathrm{itc}(\mathbf{L_2})(\mathbf{1 + m}) - \sum_{\sigma \in \mathbf{\Sigma}} \mathrm{itc}_\sigma(\mathbf{L_2})\, \mathrm{itc}_\sigma(\mathbf{L_1})$	$\mathrm{ntc}(L_1) + \mathrm{ntc}(L_2) + s(L_1) + s(L_2)$
$L_1 \cap L_2$	$\mathrm{itc}(L_1)\, \mathrm{itc}(L_2)$	$\sum_{\sigma \in \Sigma} \mathrm{ntc}_\sigma(L_1)\, \mathrm{ntc}_\sigma(L_2)$
L^C	$\|\Sigma\|(\mathrm{itc}(L) + 2)$	$\|\Sigma\| 2^{\mathrm{ntc}(L)+1}$ $2^{\frac{\mathrm{ntc}(L)}{2} - 2} - 1$
$L_1 L_2$	$\|\mathbf{\Sigma}\|(\mathbf{m+1})\mathbf{2^n} - \|\mathbf{\Sigma_c^{L_2}}\|(\mathbf{f\ 2^{n-1} + 1}) - \sum_{\sigma \in \mathbf{\Sigma_i^{L_2}}} (\mathbf{2^{u_\sigma} + f\ 2^{\mathrm{itc}_\sigma(L_2)}}) - \sum_{\sigma \in \mathbf{\Sigma_{ii}}} \mathbf{\tilde{u}_\sigma 2^{u_\sigma}} - \sum_{\sigma \in \mathbf{\Sigma_{ic}}} \mathbf{\tilde{u}_\sigma}$	$\mathrm{ntc}(L_1) + \mathrm{ntc}(L_2) + f_{in}(L_1)$
$L^\star$	$\|\mathbf{\Sigma}\|(\mathbf{2^{m-l-1} + 2^{m-1}}) + \sum_{\sigma \in \mathbf{\Sigma_i}} (\mathbf{s_\sigma - 2^{\tilde{u}_\sigma}})$	$\mathrm{ntc}(L) + f_{in}(L)$
L^R	$\|\mathbf{\Sigma}\|(\mathbf{2^m - 1})$	$\mathrm{ntc}(L) + f(L)$

Table 3: Transition complexity of basic regularity preserving operations on general regular languages.

measures can be defined, for a regular language L, considering the measure values for its minimal DFA. Thus we can use following notation, $f(L)$, $s_\sigma(L)$, $t_\sigma(L)$, $u_\sigma(L)$, and $\tilde{u}_\sigma(L)$, respectively.

3. Regular Languages

Y. Gao et al. [8] were the first to study the transition complexity of Boolean operations on regular languages based on incomplete DFAs. For the intersection and the complement, tight bounds were presented, but for the union operation the upper and lower bounds differ by a factor of two. Nevertheless, they conjectured a tight upper bound for this operation.

In this section, we continue this study by extending the analysis to the concatenation, the Kleene star and the reversal operations. For these operations tight upper bounds are given. We also give a tight upper bound for the transition complexity of the union, which refutes the conjecture presented by Y. Gao et al., as we already mentioned. We also prove that the upper bounds are maximal when $f(L)$ is minimal. This study is also done for unary regular languages.

In the Tables 2 and 3 we summarize the results of this section (in bold) as well as some known results for other descriptional complexity measures: state complexity (sc), and nondeterministic transition complexity (ntc). The last measure was studied by Domaratzki and Salomaa [7, 19], and they also used refined number of transitions for a more precise computation of the operational transition complexity. In Table 3, $s(L)$ is the minimal number of transitions leaving the initial state of any transition-minimal NFA accepting L, and $f_{in}(L)$ is the number of transitions entering the final states of any transition-minimal NFA accepting L.

At the end of the section, we present some experimental results in order to analyse the descriptional complexity measures when the referred operations are performed with uniformly random generated DFAs as operands. These experiments allow the reader to make an approximate prediction of the average-case complexity of the operations.

3.1. Union

It was shown by Y. Gao et al. [8] that

$$\mathrm{itc}(L_1 \cup L_2) \leq 2(\mathrm{itc}(L_1)\,\mathrm{itc}(L_2) + \mathrm{itc}(L_1) + \mathrm{itc}(L_2)).$$

The lower bound $\mathrm{itc}(L_1)\,\mathrm{itc}(L_2) + \mathrm{itc}(L_1) + \mathrm{itc}(L_2) - 1$ was given for particular ternary language families which state complexities are relatively prime. The authors conjectured, also, that

$$\mathrm{itc}(L_1 \cup L_2) \leq \mathrm{itc}(L_1)\,\mathrm{itc}(L_2) + \mathrm{itc}(L_1) + \mathrm{itc}(L_2),$$

when $\mathrm{itc}(L_i) \geq 2$, $i = 1, 2$.

We will present an upper bound for the state complexity and we give a new upper bound for the transition complexity of the union of two regular languages. We also present families of languages for which these upper bounds are reached, witnessing, thus, that these bounds are tight.

Following, we describe the algorithm for the union of two DFAs, based on the usual product construction, that was presented by Y. Gao et al. [8, Lemma 3.1.]. Given two incomplete DFAs $A = ([0, m[, \Sigma, \delta_A, 0, F_A)$ and $B = ([0, n[, \Sigma, \delta_B, 0, F_B)$, and considering Ω_A and Ω_B as the dead states of A and B, respectively, let $C = (([0, m[\cup \{\Omega_A\}) \times ([0, n[\cup \{\Omega_B\})), \Sigma, \delta_C, (0, 0), (F_A \times ([0, n[\cup \{\Omega_B\})) \cup (([0, m[\cup \{\Omega_A\}) \times F_B))$ be a new DFA where for $\sigma \in \Sigma$, $i \in [0, m[\cup \{\Omega_A\}$, and $j \in [0, n[\cup \{\Omega_B\}$,

$$\delta_C((i,j),\sigma) = \begin{cases} (\delta_A(i,\sigma), \delta_B(j,\sigma)), & \text{if } \delta_A(i,\sigma)\downarrow\ \wedge\ \delta_B(j,\sigma)\downarrow; \\ (\delta_A(i,\sigma), \Omega_B), & \text{if } \delta_A(i,\sigma)\downarrow\ \wedge\ \delta_B(j,\sigma)\uparrow; \\ (\Omega_A, \delta_B(j,\sigma)), & \text{if } \delta_A(i,\sigma)\uparrow\ \wedge\ \delta_B(j,\sigma)\downarrow; \\ \uparrow, & \text{otherwise.} \end{cases}$$

Note that $\delta_A(\Omega_A, \sigma)$ and $\delta_B(\Omega_B, \sigma)$ are always undefined, and the pair (Ω_A, Ω_B) never occurs in the image of δ_C. It is easy to see that DFA C accepts the language $\mathcal{L}(A) \cup \mathcal{L}(B)$. The number of states and transitions which are sufficient for any DFA C are obtained in the following theorem.

Theorem 1. *For any two regular languages L_1 and L_2 with* $\mathrm{isc}(L_1) = m$ *and* $\mathrm{isc}(L_2) = n$*, one has* $\mathrm{isc}(L_1 \cup L_2) \leq mn + m + n$ *and*

$$\mathrm{itc}(L_1 \cup L_2) \leq \mathrm{itc}(L_1)(1+n) + \mathrm{itc}(L_2)(1+m) - \sum_{\sigma \in \Sigma} \mathrm{itc}_\sigma(L_1)\,\mathrm{itc}_\sigma(L_2).$$

Proof. Let A and B be the minimal DFAs that recognize L_1 and L_2, respectively. Consider the DFA C such that $\mathcal{L}(C) = \mathcal{L}(A) \cup \mathcal{L}(B)$ and C is constructed using the algorithm described above. The result for the $\mathrm{isc}(L_1 \cup L_2)$ is given by Gao et al. in [8]. Let us prove the result for the $\mathrm{itc}(L_1 \cup L_2)$. Consider the σ-transitions of A named by α_i ($i \in [1, t_\sigma(A)]$) and the undefined σ-transitions of A named by $\bar{\alpha}_l$ ($l \in [1, u_\sigma(A)+1]$). Consider also the σ-transitions of B named by β_j ($j \in [1, t_\sigma(B)]$) and the undefined σ-transitions named by $\bar{\beta}_z$ ($z \in [1, u_\sigma(B)+1]$). We need to consider one more undefined transition in each DFA which corresponds to Ω_A and Ω_B. The σ-transitions of the DFA C accepting $L_A \cup L_B$ can only have one of the following three forms: (α_i, β_j), $(\bar{\alpha}_l, \beta_j)$, and $(\alpha_i, \bar{\beta}_z)$. Thus the DFA C has $t_\sigma(A)t_\sigma(B)$ σ-transitions of the form (α_i, β_j); $t_\sigma(A)(u_\sigma(B)+1)$ σ-transitions of the form $(\bar{\alpha}_l, \beta_j)$; and $(u_\sigma(A)+1)t_\sigma(B)$ σ-transitions of the form $(\alpha_i, \bar{\beta}_z)$. As we know that $u_\sigma(A) = m - t_\sigma(A)$ and $u_\sigma(B) = n - t_\sigma(B)$, the number of σ-transitions is

$$t_\sigma(A)t_\sigma(B) + t_\sigma(A)(n - t_\sigma(B) + 1) + t_\sigma(B)(m - t_\sigma(A) + 1).$$

Therefore, with $\mathrm{itc}_\sigma(\mathcal{L}(A)) = t_\sigma(A)$ and $\mathrm{itc}_\sigma(\mathcal{L}(B)) = t_\sigma(B)$ the inequality holds. $\square$

3.1.1. Worst-case Witnesses

In this section, we show that the upper bounds established in Theorem 1 are tight. We need to consider two cases, parametrized by the state complexities of the language operands: $m \geq 2$ and $n \geq 2$; and $m = 1$ and $n \geq 2$ (or vice versa). Note that, in this section, we consider automaton families over a binary alphabet, $\Sigma = \{a, b\}$.

Case 1*:* $m \geq 2$ *and* $n \geq 2$*.* Let $A = ([0, m[, \Sigma, \delta_A, 0, \{0\})$ with $\delta_A(m-1, a) = 0$, and $\delta_A(i, b) = i + 1,\ 1 \in [0, m-1[$; and $B = ([0, n[, \Sigma, \delta_B, 0, \{n-1\})$ with $\delta_B(i, a) = i + 1,\ i \in [0, n-1[$, and $\delta_B(i, b) = i,\ i \in [0, n[$. These minimal DFAs are represented in Fig. 1 and Fig. 2, respectively.

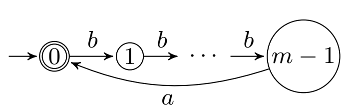

Figure 1: DFA A with m states.

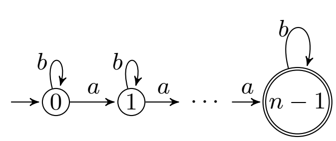

Figure 2: DFA B with n states.

Theorem 2. *For any integers $m \geq 2$ and $n \geq 2$, there exist an m-state DFA A with $r = m$ transitions and an n-state DFA B with $s = 2n - 1$ transitions such that any DFA accepting $\mathcal{L}(A) \cup \mathcal{L}(B)$ needs, at least, $mn + m + n$ states and $(r + 1)(s + 1)$ transitions.*

Proof. Let us count the number of states of the DFA C accepting $\mathcal{L}(A) \cup \mathcal{L}(B)$, constructed by the previous algorithm. Consider the pairs (i, j) representing states of that DFA C. Then for each (i, j) where $i \in ([0, m[\cup \Omega_A)$ and $j \in ([0, n[\cup \Omega_B)$ except the case when $(i, j) = (\Omega_A, \Omega_B)$, there exists a word

$$w = \begin{cases} (b^{m-1}a)^j b^i, & \text{if } i \neq \Omega_A \wedge j \neq \Omega_B; \\ (b^{m-1}a)^n b^i, & \text{if } i \neq \Omega_A \wedge j = \Omega_B; \\ b^m a^j, & \text{if } i = \Omega_A \wedge j \neq \Omega_B; \end{cases}$$

which represents each state, i.e., a different left quotient. Thus there are at least $mn + m + n$ distinct left quotients (states of C).

Let us consider the number of transitions of DFA C. If we name the defined and undefined transitions of the DFAs A and B as in the proof of the Theorem 1 then C has:

- $mn + n - m + 1$ a-transitions because there exist $n - 1$ a-transitions of the form (α_i, β_j); 2 a-transitions of the form $(\alpha_i, \bar{\beta}_j)$; and $m(n - 1)$ a-transitions of the form $(\bar{\alpha}_i, \beta_j)$;
- $mn+m+n-1$ b-transitions because there exist $(m-1)n$ b-transitions of the form (α_i, β_j); $m-1$ b-transitions of the form $(\alpha_i, \bar{\beta}_j)$; and $2n$ b-transitions of the form $(\bar{\alpha}_i, \beta_j)$.

As $r = m$ and $s = 2n - 1$, DFA C has $(r + 1)(s + 1)$ transitions. □

The referred conjecture $\text{itc}(L_1 \cup L_2) \leq \text{itc}(L_1)\,\text{itc}(L_2) + \text{itc}(L_1) + \text{itc}(L_2)$ fails for these families because, as we prove in the previous theorem, $\text{itc}(L_1 \cup L_2) = (r + 1)(s + 1)$, where $r = \text{itc}(L_1)$ and $s = \text{itc}(L_2)$, then $\text{itc}(L_1 \cup L_2) = \text{itc}(L_1)\,\text{itc}(L_2) + \text{itc}(L_1) + \text{itc}(L_2) + 1$.

Case 2: $m = 1$ and $n \geq 2$. Let $A = (\{0\}, \Sigma, \delta_A, 0, \{0\})$ with $\delta_A(0, a) = 0$, and consider the DFA B defined in the previous case.

Theorem 3. *For any integer $n \geq 2$, there exists an 1-state DFA A with one transition and an n-state DFA B with $s = 2n-1$ transitions such that any DFA accepting $\mathcal{L}(A) \cup \mathcal{L}(B)$ has, at least, $2n + 1$ states and $2(s + 1)$ transitions.*

Proof. Consider the DFA C, accepting $\mathcal{L}(A) \cup \mathcal{L}(B)$, constructed by the previous algorithm. As in the proof of Theorem 2, let us see the states of DFA C as pairs (i, j) where $i \in (\{0\} \cup \Omega_A)$ and $j \in ([0, n[\cup \Omega_B)$ except the case when $(i, j) = (\Omega_A, \Omega_B)$. For each of those pairs, there exists a word,

$$w = \begin{cases} a^j, & \text{if } i \neq \Omega_A \wedge j \neq \Omega_B; \\ ba^j, & \text{if } i = \Omega_A \wedge j \neq \Omega_B; \\ a^n, & \text{if } i \neq \Omega_A \wedge j = \Omega_B; \end{cases}$$

which represents a state of C, i.e., a different left quotient. Thus there are at least $2n+1$ distinct left quotients.

Let us consider the transitions named as in the proof of the Theorem 1, then DFA C has:

- $2n$ a-transitions because there exist $n-1$ a-transitions of the form (α_i, β_j); 2 a-transitions of the form $(\alpha_i, \bar{\beta}_j)$; and $n-1$ a-transitions of the form $(\bar{\alpha}_i, \beta_j)$;
- $2n$ b-transitions because by this symbol there are only transitions of the form $(\bar{\alpha}, \beta_j)$.

Thus, the DFA C has $4n$ transitions. As $r=1$ and $s=2n-1$, the DFA C has $2(s+1)$ transitions. Note that $r=1$ and, thus, $2(s+1)=(r+1)(s+1)$. □

3.2. Concatenation

In this section we deal with the incomplete descriptional complexity of the concatenation of two regular languages.

The construction used is as follows. Given two incomplete DFAs, $A = ([0,m[, \Sigma, \delta_A, 0, F_A)$ and $B = ([0,n[, \Sigma, \delta_B, 0, F_B)$, a DFA accepting $\mathcal{L}(A)\mathcal{L}(B)$ is $C = (R, \Sigma, \delta_C, r_0, F_C)$ where for $\sigma \in \Sigma$, $i \in [0,m[$, and $P \subseteq [0,n[$, $R \subset ([0,m[\cup \{\Omega_A\}) \times 2^{[0,n[}$ (precisely defined in the proof of Theorem 4); r_0 is $(0,\emptyset)$ if $0 \notin F_A$, and is $(0,\{0\})$ otherwise; $F_C = \{(i,P) \in R \mid P \cap F_B \neq \emptyset\}$; and

$$\delta_C((q,T),\sigma) = \begin{cases} (\delta_A(q,\sigma), \delta_B(T,\sigma) \cup \{0\}), & \text{if } \delta_A(q,\sigma)\downarrow \ \wedge \ \delta_A(q,\sigma) \in F_A; \\ (\delta_A(q,\sigma), \delta_B(T,\sigma)), & \text{if } \delta_A(q,\sigma)\downarrow \ \wedge \ \delta_A(q,\sigma) \notin F_A; \\ (\Omega_A, \delta_B(T,\sigma)). & \text{if } \delta_A(q,\sigma)\uparrow \ \wedge \ \delta_B(T,\sigma) \neq \emptyset; \\ \uparrow, & \textit{otherwise.} \end{cases}$$

In the following, we determine the number of states and transitions that are sufficient for any DFA C resulting from the previous construction.

Given an automaton A, its alphabet can be partitioned in two sets, Σ_c^A and Σ_i^A, such that $\sigma \in \Sigma_c^A$ if A is σ-complete, and $\sigma \in \Sigma_i^A$ otherwise. In the same way, considering two automata A and B, the alphabet can be divided into four disjoint sets Σ_{ci}, Σ_{cc}, Σ_{ii} and Σ_{ic}. As before, these notations can be extended to regular languages considering their minimal DFAs.

Theorem 4. *For any regular languages L_1 and L_2 with $\operatorname{isc}(L_1) = m$, $\operatorname{isc}(L_2) = n$, $u_\sigma = u_\sigma(L_2)$, $f = f(L_1)$ and $\tilde{u}_\sigma = \tilde{u}_\sigma(L_1)$, one has $\operatorname{isc}(L_1L_2) \leq (m+1)2^n - f2^{n-1} - 1$, and*

$$\begin{aligned}\operatorname{itc}(L_1L_2) \leq |\Sigma|(m+1)2^n - |\Sigma_{ic} \cup \Sigma_{cc}|(f2^{n-1}+1) - \\ - \sum_{\sigma \in (\Sigma_{ci} \cup \Sigma_{ii})} (2^{u_\sigma} + f2^{\operatorname{itc}_\sigma(L_2)}) - \sum_{\sigma \in \Sigma_{ii}} \tilde{u}_\sigma 2^{u_\sigma} - \sum_{\sigma \in \Sigma_{ic}} \tilde{u}_\sigma.\end{aligned}$$

Proof. Let A and B be the minimal DFAs that recognize L_1 and L_2, respectively. Consider the DFA C such that $\mathcal{L}(C) = \mathcal{L}(A)\mathcal{L}(B)$, constructed using the algorithm described above. First, let us consider the problem of $\text{isc}(L_1L_2)$. The set R is a set of pairs (s, P) where $s \in ([0, m[\cup\Omega_A)$, and $P \subseteq [0, n[$. There exist $(m+1)2^n$ such pairs. However, we know that R does not contain the pairs in which s is a final state of A and the set P does not contain the initial state of B. Thus, we need to remove $f(A)2^{n-1}$ pairs from the first counting. As the pair $(\Omega_A, \emptyset)$ is not in R, we can also remove it. The resulting number of states is, thus, $(m + 1)2^n - f(A)2^{n-1} - 1$.

Now, let us consider the problem of estimating $\text{itc}(L_1L_2)$. We name the σ-transitions of A and B as in the proof of the Theorem 1 with a slight modification: $z \in [1, u_\sigma(B)]$. The σ-transitions of C are pairs (θ, γ) where θ is either an α_i or an $\bar{\alpha}_l$, and γ is a set of β_j or $\bar{\beta}_z$. By construction, C cannot have transitions where θ is an $\bar{\alpha}_l$, and γ is a set with only $\bar{\beta}_k$, because these pairs would correspond to undefined transitions. If $\sigma \in \Sigma_{ci}$, the number of C σ-transitions is $(t_\sigma(A) + 1)2^{t_\sigma(B)+u_\sigma(B)} - 2^{u_\sigma(B)} - f(A)2^{t_\sigma(B)}$, because the number of θs is $t_\sigma(A) + 1$ and the number of γs is $2^{t_\sigma(B)+u_\sigma(B)}$. We need to remove the $2^{u_\sigma(B)}$ sets of transitions of the form $(v, \emptyset)$ where v corresponds to the undefined σ-transition leaving the state Ω_A. If θ corresponds to a transition that leaves a final state of A, then γ needs to include the initial state of B. Thus we also remove $f(A)2^{t_\sigma(B)}$ pairs. If $\sigma \in \Sigma_{cc}$, C has $(t_\sigma(A) + 1)2^{t_\sigma(B)} - 1 - f(A)2^{t_\sigma(B)-1}$ σ-transitions. In this case, $u_\sigma(B) = 0$. The only pair we need to remove is $(v, \emptyset)$ where v corresponds to the undefined σ-transition leaving the state Ω_A. Analogously, if $\sigma \in \Sigma_{ii}$, C has $(t_\sigma(A)+u_\sigma(A)+1)2^{t_\sigma(B)+u_\sigma(B)} - (\tilde{u}_\sigma(A)+1)2^{u_\sigma(B)} - f(A)2^{t_\sigma(B)}$ σ-transitions. Finally, if $\sigma \in \Sigma_{ic}$, C has $(t_\sigma(A)+u_\sigma(A)+1)2^{t_\sigma(B)} - (\tilde{u}_\sigma(A)+1) - f(A)2^{t_\sigma(B)-1}$ σ-transitions. Thus, after some simplifications, the right side of the inequality in the proposition holds. □

Corollary 1. *The* $\text{isc}(L_1L_2)$ *in the Theorem 4 is maximal when* $f(L_1) = 1$.

3.2.1. Worst-case Witnesses

In the following we show that the complexity upper bounds found in Theorem 4 are tight. As in Section 3.1.1, we need to consider three different cases, according to the state and transition complexities of the operands. Although the tight bound for (complete) state complexity can be reached over a binary alphabet [14], all automaton families used in this section have an alphabet $\Sigma = \{a, b, c\}$.

Case 1*:* $m \geq 2$ *and* $n \geq 2$. Let $A = ([0, m[, \Sigma, \delta_A, 0, \{m-1\})$ with $\delta_A(i, a) = i+1 \bmod m$, if $i \in [0, m[$, $\delta_A(i, b) = 0$, if $i \in [1, m[$, and $\delta_A(i, c) = i$ if $i \in [0, m[$; and $B = ([0, n[, \Sigma, \delta_B, 0, \{n-1\})$ with $\delta_B(i, a) = i$ if $i \in [0, n[$, $\delta_B(i, b) = i+1 \bmod n$, if $i \in [0, n[$, and $\delta_B(i, c) = 1,\ i \in [1, n[$. These automata are simple modifications of the ones presented in the proof of the Theorem 2.1 in [26]: a b-transition from the state 0 to itself on DFA A, and a c-transition from the state 0 to the state 1 were eliminated. Both automata are represented in Fig. 3.

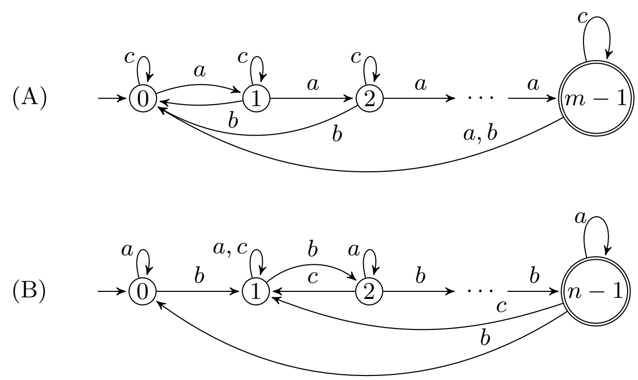

Figure 3: DFA A with m states and DFA B with n states.

Theorem 5. *For any integers $m \geq 2$ and $n \geq 2$, there exist an m-state DFA A with $r = 3m - 1$ transitions and an n-state DFA B with $s = 3n - 1$ transitions such that any DFA accepting $\mathcal{L}(A)\mathcal{L}(B)$ has, at least, $(m+1)2^n - 2^{n-1} - 1$ states and $(r+1)2^{\frac{s+1}{3}} + 3.2^{\frac{s-2}{3}} - 5$ transitions.*

Proof. Consider the DFA C such that $\mathcal{L}(C) = \mathcal{L}(A)\mathcal{L}(B)$ and C is constructed using the concatenation algorithm described above. First we prove the result for the number of states, following the proof of the Theorem 2.1 in [26]. From each $w \in \{a, b\}^\star$, let $S(w) = \{ \ i \mid w = w'w''$ such that $w' \in \mathcal{L}(A)$ *and* $i = |w''|_b \mod n \ \}$, where $|w|_b$ denotes the number of occurrences of the symbol b in the word w. Consider $w, w' \in \{a, b\}^\star$ such that $S(w) \neq S(w')$. Let $k \in S(w) \setminus S(w')$ (or $S(w') \setminus S(w)$). It is clear that $wb^{n-1-k} \in \mathcal{L}(A)\mathcal{L}(B)$ but $w'b^{n-1-k} \notin \mathcal{L}(A)\mathcal{L}(B)$.

For each $w \in \{a, b\}^\star$, define $T(w) = max\{ \ |w''| \mid w = w'w''$ *and* $w'' \in a^\star \ \}$. Consider $w, w' \in \{a, b\}^\star$ such that $S(w) = S(w')$ and $T(w) > T(w') \mod m$. Let $i = T(w) \mod m$ and $w'' = a^{m-1-i}b^{n-1}$. Therefore $ww'' \in \mathcal{L}(A)\mathcal{L}(B)$, but $w'w'' \notin \mathcal{L}(A)\mathcal{L}(B)$ because it has at least less one a than ww''.

For each subset $s = \{i_1, \ldots, i_t\} \subseteq [0, n[$, where $i_1 > \cdots > i_t$, and an integer $j \in [0, \ldots, m[\cup \{\Omega_A\}$ except the cases where $0 \notin s$ and $j = m - 1$, and $s = \emptyset$ and $j = \Omega_B$, there exists a word

$$w = \begin{cases} a^{m-1}b^{i_1} \cdots a^{m-1}b^{i_t}a^j, & \text{if } j \neq \Omega_A; \\ a^{m-1}b^{i_1} \cdots a^{m-1}b^{i_t}b^n, & \text{if } j = \Omega_A; \end{cases}$$

such that $S(w) = s$ and $T(w) = j$, which represents a different left quotient induced by $\mathcal{L}(A)\mathcal{L}(B)$. Thus, C is minimal and has $(m + 1)2^n - 2^{n-1} - 1$ states.

Considering, now, the number of transitions. As in the proof of Theorem 4, the transitions of C are pairs (θ, γ). Then, C has:

- $(m+1)2^n - 2^{n-1} - 1$, a-transitions. There are $m+1$ θs and 2^n γs, from which we need to remove the transition $(\Omega_A, \emptyset)$. If θ is a transition which leaves a final state of A, γ needs to include the transition that leaves the initial state of B. Thus, 2^{n-1} pairs are removed.
- $(m+1)2^n - 2^{n-1} - 2$, b-transitions. Here, the transition $(\bar{\theta}, \emptyset)$ is removed.
- $(m+1)2^n - 2^{n-1} - 2$, c-transitions. This is analogous to the previous case.

As $m = \frac{r+1}{3}$ and $n = \frac{s+1}{3}$, the DFA C has $(r+1)2^{\frac{s+1}{3}} + 3.2^{\frac{s-2}{3}} - 5$ transitions. $\square$

Case 2: $m = 1$ and $n \geq 2$. Let $A = (\{0\}, \Sigma, \delta_A, 0, \{0\})$ with $\delta_A(0, b) = \delta_A(0, c) = 0$; and $B = ([0, n[, \Sigma, \delta_B, 0, \{n-1\})$ with $\delta_B(i, a) = i$ if $i \in [0, n[$, $\delta_B(i, b) = i + 1 \bmod n$ if $i \in [0, n[$, and $\delta_B(i, c) = i + 1 \bmod n$, if $i \in [1, n[$. The automata A and B are represented in Fig. 4.

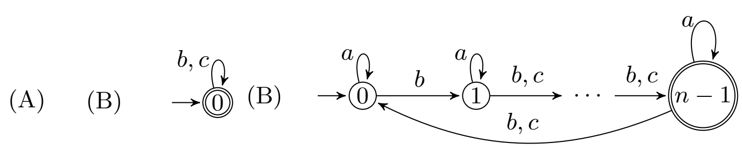

Figure 4: DFA A with 1 state and DFA B with n states.

Theorem 6. *For any integer $n \geq 2$, there exist a 1-state DFA A with 2 transitions and an n-state DFA B with $s = 3n - 1$ transitions such that any DFA accepting $\mathcal{L}(A)\mathcal{L}(B)$ has, at least, $2^{n+1} - 2^{n-1} - 1$ states and $3(2^{\frac{s+4}{3}} - 2^{\frac{s-2}{3}}) - 4$ transitions.*

Proof. Consider the DFA $C = (R, \Sigma, \delta, 0, F)$, constructed by the concatenation algorithm previously defined, such that $\mathcal{L}(C) = \mathcal{L}(A)\mathcal{L}(B)$. One needs to prove that C is minimal, i.e. all states are reachable from the initial state and are pairwise distinguishable. The automaton C has states (q, P) with $q \in \{\Omega_A, 0\}$, $P = \{i_1, \ldots, i_k\}$, $1 \leq k \leq n$, and $i_1 < \cdots < i_k$. There are two kinds of states: final states where $i_k = n - 1$; and non-final states where $i_k \neq n - 1$. Note that, whenever $q = 0$, we have $i_1 = 0$.

Let f be a final state of the form (q, P), where $P = \{i_1, \ldots, i_{k-1}, n-1\}$ and $\bar{P} = [0, n[\setminus P$. Let us construct a word w of size n, such that $\delta(0, w) = f$. We will count the positions (starting with zero) of the word w from the last to the first. If f has $q = \Omega_A$, w has an a in the position i_1; c's in the positions $j \in \bar{P} \setminus \{i_1 - 1\}$ if $i_1 \neq 0$, or $j \in \bar{P}$ otherwise; all the other positions are b's. For example, if $n = 5$, $P = \{4\}$ and $\bar{P} = \{0, 1, 2, 3\}$ then $w = abccc$. If f has $q = 0$ the word has c's in all positions $i_j - 1$, $i_j \in \bar{P}$; all the other positions are $b's$. For example, if $P = \{0, 4\}$, $\bar{P} = \{1, 2, 3\}$ and $n = 5$ then $w = bbccc$. Now, consider the non-final states p which have the same form (q, P), but $i_k \neq n - 1$ and $\bar{P} = \{0, \ldots, n-2\} \setminus P$. The word w for these non-final states is constructed with the same rules described above for final states. This proves that all states are reachable from initial state.

Now let us prove that all states are pairwise distinguishable. Final states are trivially distinguishable from non-final states. We need to prove that states of the same kind are distinguishable. Consider $w, w' \in \Sigma^\star$ such that $\delta(0, w) = q$ and $\delta(0, w') = p$, $q \neq p$. Suppose that q and p are final. There are three cases to consider. Let $q = (0, \{0, i_2, \ldots, i_k, n-1\})$ and $p = (0, \{0, j_2, \ldots, j_{k'}, n-1\})$. Suppose $k \geq k'$ and $i \in \{0, i_2, \ldots, i_k, n-1\} \setminus \{0, j_2, \ldots, j_{k'}, n-1\}$. Then $wc^{n-1-i} \in \mathcal{L}(C)$ but $w'c^{n-1-i} \notin \mathcal{L}(C)$. If $q = (\Omega_A, \{i_1, \ldots, i_k, n-1\})$ and $p = (\Omega_A, \{j_1, \ldots, j_{k'}, n-1\})$, we can take i as before and then $wb^{n-1-i} \in \mathcal{L}(C)$ but $w'b^{n-1-i} \notin \mathcal{L}(C)$. If $q = (0, \{0, i_2, \ldots, i_k, n-1\})$ and $p = (\Omega_A, \{j_1, \ldots, j_{k'}, n-1\})$, then $wc^n b^{n-1} \in \mathcal{L}(C)$ but $w'c^n b^{n-1} \notin \mathcal{L}(C)$. Now suppose that q and p are non-final. Let $q = (0, \{0, i_2, \ldots, i_k\})$ and $p = (0, \{0, j_2, \ldots, j_{k'}\})$. Consider, without loss of generality, $k \geq k'$ and $i \in \{0, i_2, \ldots, i_k\} \setminus \{0, j_2, \ldots, j_{k'}\}$. It is clear that $wc^{n-1-i} \in \mathcal{L}(C)$ but $w'c^{n-1-i} \notin \mathcal{L}(C)$. If $q = (\Omega_A, \{i_1, \ldots, i_k\})$ and $p = (\Omega_A, \{j_1, \ldots, j_{k'}\})$, we can take $i \in \{i_1, \ldots, i_k\} \setminus \{j_1, \ldots, j_{k'}\}$ and then $wb^{n-1-i} \in \mathcal{L}(C)$ but $w'b^{n-1-i} \notin \mathcal{L}(C')$. Finally, if $q = (0, \{0, i_2, \ldots, i_k\})$ and $p = (\Omega_A, \{j_1, \ldots, j_{k'}\})$, clearly $wc^n b^{n-1} \in \mathcal{L}(C)$ but $w'c^n b^{n-1} \notin \mathcal{L}(C)$. Thus C is minimal and has $2^{n-2} + 2^{n-1}$ final states and $2^{n-2} + 2^{n-1} - 1$ non-final states. Therefore, it has $2^{n+1} - 2^{n-1} - 1$ states.

The proof for the number of transitions is similar to the proof for the number of transitions of Theorem 5. □

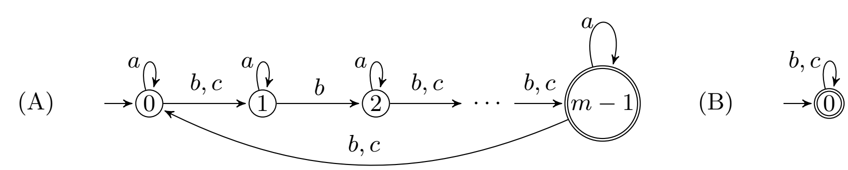

Figure 5: DFA A with m states and DFA B with 1 state.

Case 3: $m \geq 2$ *and* $n = 1$. Let $A = ([0, m[, \Sigma, \delta_A, 0, \{m-1\})$ with $\delta_A(i, a) = i$, if $i \in [0, m[$, $\delta_A(i, b) = i + 1 \bmod m$, if $i \in [0, m[$, $\delta_A(i, c) = i + 1 \bmod m$ if $i \in [0, m[\setminus[1]$; and $B = (\{0\}, \Sigma, \delta_B, 0, \{0\})$ with $\delta_B(0, b) = \delta_B(0, c) = 0$. A representation of these DFAs can be seen in Fig. 5.

Theorem 7. *For any integer $m \geq 2$, there exist an m-state DFA A with $r = 3m - 1$ transitions and an 1-state DFA B with 2 transitions such that any DFA accepting $\mathcal{L}(A)\mathcal{L}(B)$ has at least $2m$ states and $2r$ transitions.*

Proof. Consider the DFA $C = (Q, \Sigma, \delta, 0, F)$, such that $\mathcal{L}(C) = \mathcal{L}(A)\mathcal{L}(B)$, constructed with the previous algorithm. We only present the proof for the number of states because the proof for the number of transitions is similar to the proof of Theorem 5. By construction we know that C has two kinds of p states:

- final states, which are of the form $(x, \{0\})$ where $x \in [0, m[\cup \{\Omega_A\}$.
- non-final states, which are of the form $(x, \emptyset)$ where $x \in [0, m-2]$.

For any state p we can find a word w for which $\delta(0, w) = p$. If p is a final state of the form $(x, \{0\})$ where $x \in [0, m[$ then $w = b^{m+x}$. In case $x = \Omega_A$ then $w = b^{m+1}c$. Finally, if p is a non-final state then $w = b^x$. Thus, all states are reachable from the initial state. Let us prove that the final states are distinguishable:

- The final states where $x \in [0, m[$ are not equivalent because they correspond to the states of the DFA A which is minimal.
- The final state where $x = \Omega_A$ is not equivalent to the other final state because it is the only final state which is σ-incomplete.

Let $(i, \emptyset)$ and $(j, \emptyset)$ be two distinct non-final states. Consider $w_i, w_j \in \Sigma^\star$ such that $\delta_C(r_0, w_i) = (i, \emptyset)$ and $\delta_C(r_0, w_j) = (j, \emptyset)$. It is clear that $w_i a^{i+1} b^{m-1-i} a^{i+1}$ belongs to $\mathcal{L}(A)\mathcal{L}(B)$ but $w_j a^{i+1} b^{m-1-i} a^{i+1}$ does not. Then w_i and w_j are in different left quotients induced by $\mathcal{L}(A)\mathcal{L}(B)$. Hence, the DFA C is minimal and has $2m$ states. $\square$

3.3. Kleene Star

In this section we give a tight upper bound for the incomplete transition complexity of the star operation. The incomplete state complexity of this operation coincides with the one for the complete case.

Let $A = ([0, n[\,, \Sigma, \delta, 0, F)$ be a DFA. Consider $F_0 = F \setminus \{0\}$ and suppose that $l = |F_0| \geq 1$. If $F = \{0\}$, then $\mathcal{L}(A)^\star = \mathcal{L}(A)$. The following algorithm constructs a DFA for the Kleene star of A. Let $A' = (Q', \Sigma, \delta', q_0', F')$ be a new DFA where $q_0' \notin Q$ is a new initial state, $Q' = \{q_0'\} \cup \{P \mid P \subseteq (Q \setminus F_0) \wedge P \neq \emptyset\} \cup \{P \mid P \subseteq Q \wedge 0 \in P \wedge P \cap F_0 \neq \emptyset\}$, $F' = \{q_0'\} \cup \{R \mid R \subseteq Q \wedge R \cap F \neq \emptyset\}$, and for $\sigma \in \Sigma$,

$$\delta'(q_0', \sigma) = \begin{cases} \{\delta(0, \sigma)\}, & \text{if } \delta(0, \sigma)\downarrow \wedge\, \delta(0, \sigma) \notin F_0; \\ \{\delta(0, \sigma), 0\}, & \text{if } \delta(0, \sigma)\downarrow \wedge\, \delta(0, \sigma) \in F_0; \\ \emptyset, & \text{if } \delta(0, \sigma)\uparrow; \end{cases}$$

and

$$\delta'(R, \sigma) = \begin{cases} \delta(R, \sigma), & \text{if } \delta(R, \sigma) \cap F_0 = \emptyset; \\ \delta(R, \sigma) \cup \{0\}, & \text{if } \delta(R, \sigma) \cap F_0 \neq \emptyset; \\ \emptyset, & \text{if } \delta(R, \sigma) = \emptyset. \end{cases}$$

It is easy to verify that A' recognizes the language $\mathcal{L}(A)^\star$. In the following we present the upper bounds for the number of states and transitions for any DFA A' resulting from the algorithm described above.

Theorem 8. *For any regular language L, with $\mathrm{isc}(L) = n$, $s_\sigma = s_\sigma(L)$, one has $\mathrm{isc}(L^\star) \leq 2^{n-1} + 2^{n-l-1}$ and $\mathrm{itc}(L^\star) \leq |\Sigma|(2^{n-1} + 2^{n-l-1}) + \sum_{\sigma \in \Sigma_i} (s_\sigma - 2^{\tilde{u}_\sigma})$.*

Proof. Let A be the minimal DFA that recognizes L. Consider the DFA A' such that $\mathcal{L}(A') = \mathcal{L}(A^\star)$ and A' is constructed using the algorithm defined above. Let us prove the result for the $\mathrm{isc}(L^\star)$. Note that Q' is defined as the union of three different sets. The first set contains only the initial state. The states generated by the second set of Q' are the non-empty parts of Q disjoint from F_0. So in this set we have $2^{n-l} - 1$ states (we also remove the empty set). The states in the third set of Q' are the parts of Q that contains the initial state of A and are non-disjoint from F_0. Those are at most $(2^l - 1)2^{n-l-1}$. Therefore the number of states is lesser or equal than $2^{n-1} + 2^{n-l-1}$.

Let us consider the $\mathrm{itc}(L^\star)$. Following the analysis done for the states, the number of σ-transitions of A' is the summation of:

1. s_σ σ-transitions leaving the initial state of A.
2. the number of sets of σ-transitions leaving only non-final states of A:
   (a) $(2^{t_\sigma - l}) - 1$, if A is σ-complete, because we have $t_\sigma - l$ σ-transitions of this kind, and we remove the empty set;
   (b) $2^{t_\sigma - l + u_\sigma} - 2^{\tilde{u}_\sigma}$, if A is σ-incomplete because we have $t_\sigma - l + u_\sigma$ of this kind, and we subtract the number of sets with only undefined σ-transitions of A.
3. the number of sets of σ-transitions leaving final and non-final states of A. We do not count the transition leaving the initial state of A because, by construction, if a transition of A' contains a transition leaving a final state of A then it also contains the one leaving the initial state of A. Thus, we have
   (a) $(2^l - 1)2^{t_\sigma - l - 1}$, if A is σ-complete;
   (b) $(2^l - 1)2^{t_\sigma - l - 1 + u_\sigma}$, if A is σ-incomplete.

Thus, the inequality in the proposition holds. □

Corollary 2. *The* $\mathrm{isc}(L^\star)$ *presented in Theorem 8 is maximal when* $l = 1$.

3.3.1. Worst-case Witnesses

Let us present an automaton family, with $\Sigma = \{a, b\}$, for which the upper bounds in Theorem 8 are reached.

Define $A = ([0, n[, \Sigma, \delta_A, 0, \{n-1\})$ with $\delta_A(i, a) = i + 1 \bmod n$ for $i \in [0, n[$, and $\delta_A(i, b) = i + 1 \bmod n$ for $i \in [1, n[$. This DFA is depicted in Fig.6.

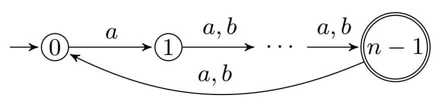

Figure 6: DFA A with n states.

Theorem 9. *For any integer $n \geq 2$, there exist an n-state DFA A with $r = 2n-1$ transitions such that any DFA accepting $\mathcal{L}(A)^\star$ has, at least, $2^{n-1}+2^{n-2}$ states and $2^{\frac{r+1}{2}} + 2^{\frac{r-1}{2}} - 2$ transitions.*

Proof. For $n = 2$ it is clear that $\mathcal{L} = \{w \in \{a,b\}^\star \mid |w|_a \textit{ is odd}\}$ is accepted by a two-state DFA, and $\mathcal{L}^\star = \{\varepsilon\} \cap \{w \in \{a,b\}^\star \mid |w|_a \geq 1\}$ cannot be accepted with less than 3 states. For $n > 2$, we consider the automaton family A which is shown in Fig. 6. Consider the DFA A' such that $\mathcal{L}(A') = \mathcal{L}(A^\star)$. First we prove the result for the number of states, following the proof of the Theorem 3.3 in [26]. In order to prove that A' is minimal, thus we need to prove the following.

- Every state is reachable from the start state. As each state of A' is a subset of states of A, we proceed by induction on the size of these states. If $|q| = 1$ we have:

$$q = \begin{cases} \{1\} = \delta'(q'_0, a); & (1) \\ \{i\} = \delta'(\{i-1\}, a), \qquad \text{for } 1 < i < n-1; & (2) \\ \{0\} = \delta'(\{n-1, 0\}, b). & (3) \end{cases}$$

  Note that we reach $q = \{0\}$ from a state with size two, but we reach the state $\{n-1, 0\}$ by $\delta'(\{n-2\}, a)$ and $\{n-2\}$ is already considered in (2). Thus we can reach all states such that $|q| = 1$. Now, assume that, for every state q, if $|q| < m$ then q is reachable. Let us prove that if $|q| = m$ then it is also reachable. Consider $q = \{i_1, i_2, \ldots, i_m\}$ such that $0 \leq i_1 < i_2 < \cdots < i_m < n-1$ if $n-1 \notin q$, $0 = i_1 < i_2 < \cdots < i_{m-1} < i_m = n-1$ otherwise. There are three cases to consider:

  (i) $\{n-1, 0, i_3, \ldots, i_m\} = \delta'(\{n-2, i_3-1, \ldots, i_m-1\}, a)$ where the state $\{n-2, i_3-1, \ldots, i_m-1\}$ contains $m-1$ states.

  (ii) $\{0, 1, i_3, \ldots, i_m\} = \delta'(\{n-1, 0, i_3-1, \ldots, i_m-1\}, a)$ where the state $\{n-1, 0, i_3-1, \ldots, i_m-1\}$ is considered in case (i).

  (iii) $\{t, i_2, \ldots, i_m\} = \delta'(\{0, i_2-t, \ldots, i_m-t\}, a^t)$, $t > 0$, where the state $\{0, i_2-t, \ldots, i_m-t\}$ is considered in case (ii).

- Each state defines a different left quotient induced by $\mathcal{L}(A')$. Consider p, $q \in Q'$, $p \neq q$ and $i \in p \backslash q$. Then $\delta'(p, a^{n-1-i}) \in F'$ but $\delta'(q, a^{n-1-i}) \notin F'$.

Let us consider, now, the problem of the number of transitions. The DFA A' has:

- $2^{n-1} + 2^{n-2}$ a–transitions because it has one a–transition which corresponds to s_a, $2^{n-1} - 1$ a–transitions which corresponds to case 2. of Theorem 8 and 2^{n-2} a–transitions which corresponds to case 3. of Theorem 8.

- $2^{n-1} - 2 + 2^{n-2}$ b–transitions because it has $2^{n-2+1} - 2$ b–transitions which corresponds to case 2. of Theorem 8, and 2^{n-3+1} b–transitions which corresponds to case 3. of Theorem 8.

As $n = \frac{r+1}{2}$, A' has $2^{\frac{r+1}{2}} + 2^{\frac{r-1}{2}} - 2$ transitions. $\square$

3.4. Reversal

It is known that when considering complete DFAs the state complexity of the reversal operation reaches the upper bound 2^n, where n is the state complexity of the operand language. By the subset construction, a (complete) DFA resulting from the reversal has a state which corresponds to $\emptyset$, which is a dead state. Therefore, if we remove that state the resulting automaton is not complete and the incomplete state complexity is $2^n - 1$. Consequently the transition complexity is $|\Sigma|(2^n - 1)$. It is easy to see that the worst case of the reversal operation is reached when the operand is complete.

3.5. Unary Languages

In the case of unary languages, if a DFA is not complete it represents a finite language. Thus, the worst-case state complexity of operations occurs when the operand DFAs are complete. For these languages the (incomplete) transition complexity coincide with the (incomplete) state complexity. The study for union and intersection was made by Y. Gao et al. [8], and using similar methods, it is not difficult to obtain the corresponding results for the other operations addressed in this article.

3.6. Experimental Results

Hitherto we studied the descriptional complexity of several operations considering the worst-case analysis. However, for practical applications, it is important to know how significant are these worst-case results, i.e. if these upper bounds are reached for a significant number of cases or, on the contrary, only rarely occur. To evaluate this, we performed some experimental tests in order to analyse how often the upper bounds were, in practice, achieved. Although we fixed the size of the alphabet and consider small values of n and m, the experiments are statistically significant and provide valuable information about the average case behaviour of these operations.

Almeida et al. [1] presented an uniform random generator for complete DFAs. We can use this generator to obtain incomplete DFAs, if we consider the existence of a dead state. However, in this case, the probability that a state has a transition to the dead state is $\frac{1}{n+1}$, where n is the number of useful states of the generated incomplete DFA. Although this corresponds to a uniform distribution, for very large values of n, the referred probability is very low, and thus the generated DFAs are almost always complete. Therefore, in order to generate random incomplete DFAs, we can increase the number of void transitions in the generated DFAs to change the referred probability. For that, the generator accepts a parameter b that defines the multiplicity of dead states. Using b $(0 < b < 1)$, we compute the integer part of $m = \frac{b \times n}{1-b}$, which indicates the number of dead states in the generated DFA. Note that the generated DFA becomes "more incomplete" when b tends to 1.

All the tests were performed using the random generator described above. The tests and the generator were implemented in Python[2] using the FAdo system, and are both publicly available[3]. In the following experiments (Table 4) we consider $b = 0.7$.

As the DFAs were obtained with a uniform random generator, the size of each sample (20000 elements) is sufficient to ensure a 95% confidence level within a 1% error margin. Table 4 shows the results of experimental tests with 20000 pairs of incomplete DFAs as operands. We present the results for operands with $m, n \in \{2, 4, 6, 8, 10, 12, 14, 16, 18\}$ states, such that $m + n = 20$, over an alphabet of $k = 5$ symbols. As union and intersection are symmetric operations, we only present the results for $m \in \{10, 12, 14, 16, 18\}$ and $n \in \{10, 8, 6, 4, 2\}$. We considered the following measures for the DFA resulting from the operation: the state and transition complexity, sc and tc, respectively; the upper bounds for these measures, ubsc and ubtc, respectively; its transition density $d = \frac{\mathrm{tc}}{k \cdot \mathrm{sc}}$; and the ratios $\mathrm{rs} = \frac{\mathrm{sc}}{\mathrm{ubsc}}$ and $\mathrm{rt} = \frac{\mathrm{tc}}{\mathrm{ubtc}}$. Note that the results presented in this table are averages, i.e. we calculate all the referred measures for each pair of operands and then we compute the average of each measure. The columns labeled m_1, m_2, m_3 and m_4 give the maximal values of sc, ubsc, tc and ubtc, respectively. For example, considering $m = 10$ and $n = 10$ we calculate the ubsc for the concatenation of each pair of random incomplete DFAs. Then we do the average of the 20000 obtained values and the result is 8557.90, as we can see in the table. We need to do this because every measure depends of parameters that can be different in each pair of generated DFAs.

As it was expected, for the complement operation, the upper bound for the state complexity was always reached on the experiments. For all the other operations the number of states of the DFA obtained during the experimentation (sc) was much lower than the upper bounds. For example, for $m = 10$ and $n = 10$ the upper bound was 150 times larger than the number of states of the DFA resulting from the concatenation in the experiment. Even the largest DFA obtained during the experimentation has less states than what was expected in the worst case. Considering the same example, the largest DFA has 295 states and the upper bound is 8557.90. Nevertheless, for binary operations, whenever the difference between m and n increase, the number of states of the DFA resulting from the operations, in the experiment, was closer to the upper bound. For Kleene star and reversal operations, the upper bound was far from being reached. For $m = 18$ the upper bound for Kleene star was 1900 times larger than the number of states of the resulting DFA. Note that, the DFAs resulting from all the operations in the experimentation (excluding the complement) were also incomplete.

The experimental results for the transition complexity were very similar to

[2] http://www.python.org

[3] The code used to performed the tests and a more complete set of results are available at `http://khilas.dcc.fc.up.pt/~eva/` and the necessary library to perform the tests, including the referred DFA generator, can be obtained at http://fado.dcc.fc.up.pt.

b=0.7												
Concatenation												
m	n		ubsc	rs	m_1	m_2	tc	ubtc	rt	m_3	m_4	d
2	18	54.2	604404.76	0.00009	416	655359	182.90	3141223.07	0.00006	1929	3792832	0.62
4	16	55.85	253077.73	0.0002	430	294911	190.69	1316341.55	0.0001	1962	1533056	0.64
6	14	59.81	88087.17	0.0007	303	106495	210.30	468266.14	0.0004	1377	537856	0.67
8	12	59.11	28115.99	0.002	431	34815	210.50	151521.51	0.001	1928	173280	0.68
10	10	54.79	8557.90	0.01	295	10751	194.56	46208.83	0.004	1378	53300	0.68
12	8	50.72	2523.72	0.02	219	3199	180.17	13481.26	0.01	1001	15568	0.69
14	6	44.73	725.28	0.06	179	927	156.79	3760.56	0.04	750	4336	0.68
16	4	36.35	204.44	0.18	117	263	121.18	1002.60	0.12	481	1171	0.65
18	2	28.16	56.31	0.50	54	71	88.10	250.02	0.35	231	289	0.62
Union												
10	10	33.08	120	0.28	89	120	90.46	378.97	0.24	346	480	0.53
12	8	33.33	116	0.29	89	116	91.87	367.46	0.25	323	463	0.53
14	6	32.38	104	0.31	90	104	88.74	326.77	0.27	336	414	0.53
16	4	29.96	84	0.36	73	84	79.87	255.68	0.31	283	340	0.52
18	2	27.84	56	0.50	55	56	73.23	162.12	0.45	209	225	0.51
Intersection												
10	10	7.98	100	0.08	59	100	9.74	46208.83	0.0002	120	53300	0.19
12	8	8.18	96	0.09	60	96	10.09	445.26	0.02	109	825	0.19
14	6	7.78	84	0.09	56	84	9.58	389.08	0.02	101	722	0.18
16	4	6.61	64	0.10	52	64	7.93	283.61	0.03	99	624	0.17
18	2	6.03	36	0.17	34	36	7.45	155.84	0.05	70	396	0.17
Star												
2		2.07	3.23	0.64	3	4	5.22	8.73	0.60	15	19	0.50
4		4.64	10.72	0.43	12	16	13.96	40.72	0.34	51	74	0.55
6		8.79	38.20	0.23	31	64	30.55	170.63	0.18	136	302	0.68
8		14.39	141.73	0.10	74	256	53.93	676.34	0.08	333	1219	0.73
10		21.61	542.92	0.040	113	1024	85.40	2662.98	0.03	493	4987	0.77
12		30.98	2118.42	0.015	156	4096	127.16	10510.73	0.01	723	19620	0.80
14		41.10	8346.26	0.005	226	12288	173.13	41603.90	0.004	1115	60981	0.83
16		53.20	33113.56	0.002	263	49152	228.74	165364.25	0.001	1209	244731	0.85
18		68.04	131851.28	0.001	304	196608	298.15	658938.51	0.0004	1466	974212	0.87
Reversal												
2		2.43	3	0.81	3	3	5.28	15	0.35	13	15	0.42
4		6.46	15	0.43	15	15	16.48	75	0.22	63	75	0.49
6		12.18	63	0.19	48	63	34.63	315	0.11	181	315	0.54
8		18.72	255	0.07	105	255	55.43	1275	0.043	468	1275	0.56
10		26.46	1023	0.0259	129	1023	80.79	5115	0.0158	536	5115	0.58
12		36.08	4095	0.0088	187	4095	113.74	20475	0.0056	804	20475	0.60
14		45.94	16383	0.0028	224	16383	146.93	81915	0.0018	989	81915	0.61
16		57.05	65535	0.0009	353	65535	185.02	327675	0.0006	1504	327675	0.62
18		70.55	262143	0.0003	337	262143	232.92	1310715	0.0002	1476	1310715	0.63
Complement												
2		3	3	1	3	3	9.62	29.79	0.32	15	55	0.64
4		5	5	1	5	5	22.11	50.81	0.44	25	85	0.88
6		7	8	1	7	7	33.91	73.84	0.46	35	120	0.97
8		9	9	1	9	9	44.61	95.35	0.47	45	155	0.99
10		11	11	1	11	11	54.87	116.90	0.47	55	175	1.00
12		13	13	1	13	13	64.94	140.16	0.46	65	205	1.00
14		15	15	1	15	15	74.99	162.05	0.46	75	235	1.00
16		17	17	1	17	17	85	183.96	0.46	85	265	1.00
18		19	19	1	19	19	95	207.13	0.46	95	280	1.00

Table 4: Experimental results for general regular languages with $b = 0.7$

the previous ones. For the union, the difference was not so notorious, but for all the other operations it was very high, mainly for the Kleene star and the reversal operations. For example, considering $m = 10$ and $n = 10$, for union, the upper bound was only 4 times larger than the number of transitions of the resulting DFA. However, for the concatenation, the upper bound was 1300 times larger. For $m = 18$ the upper bound for reversal was 5600 times larger than the number of transitions of the resulting DFA. Note that, although the DFA resulting from the complement was complete, the upper bound for the transition complexity was much higher than the number of transitions of that DFA. This happens because Gao et al. chose to give an upper bound as a function of the transition complexity of the operand, and because of this the upper bound, in some situations, is greater than the $|\Sigma|(m+1)$, which is the maximal number of transitions of any DFA with $m+1$ states.

Although this sample was made for few values of n and m, we expect that the experimental results for other cases would be very similar. Thus, we can conjecture that the upper bounds for all operations studied are excessively pessimistic, when considering practical applications.

4. Finite Languages

For finite languages, Salomaa and Yu [20] showed that the state complexity of the determinization of an NFA with m states and k symbols is $\Theta(k^{\frac{m}{1+\log k}})$ (lower than 2^m as it is the case for general regular languages). Câmpeanu *et al.* [5] studied the operational state complexity of concatenation, Kleene star, and reversal. Finally, Han and Salomaa [9] gave tight upper bounds for the state complexity of union and intersection on finite languages.

In this section we give tight upper bounds for the state and transition complexity of all the operations considered in the last section, for incomplete DFAs representing finite languages, with an alphabet size greater than 1. Note that, for unary finite languages the incomplete transition complexity is equal to the incomplete state complexity of that language, which is always equal to the state complexity of the language minus one. For the concatenation, we correct the upper bound for the state complexity of complete DFAs [5], and show that if the *right* automaton is larger than the *left* one, the upper bound is only reached using an alphabet of variable size. In the Tables 5 and 6 we summarize the results of these section and the tight upper bounds for the state complexity on complete DFAs. As in the previous section, we also present some experimental results in order to compare the worst case with the average case for these operations.

Let A be a minimal DFA with n states accepting a finite language, where the states are assumed to be topologically ordered, *i.e.*, $p' = \delta(p, \sigma)$ implies that $p' > p$. We will denote by $\mathrm{in}_\sigma(A, i)$ the number of transitions reaching i, and omit argument A whenever there is no ambiguity. Then, $\sum_{\sigma\in\Sigma} \mathrm{in}_\sigma(0) = 0$ and there is exactly one final state which, because of the topological order is $n-1$, called *pre-dead*, such that $\sum_{\sigma\in\Sigma} t_\sigma(n-1) = 0$. The *level* of a state i is the length of the shortest path from the initial state to i which never exceeds $n-1$.

Operation	isc	sc
$L_1 \cup L_2$	$mn - 2$	$mn - (m+n)$
$L_1 \cap L_2$	$mn - 2m - 2n + 6$	$mn - 3(m+n) + 12$
L^C	$m + 1$	m
$L_1 L_2$	$\sum\limits_{i=0}^{m-1} \min\left\{k^i, \sum\limits_{j=0}^{f(A,i)} \binom{n-1}{j}\right\} + \sum\limits_{j=0}^{f(A)} \binom{n-1}{j} - 1$	$\sum\limits_{i=0}^{m-2} \min\left\{k^i, \sum\limits_{j=0}^{f(A,i)} \binom{n-2}{j}\right\} + \sum\limits_{j=0}^{f(A)} \binom{n-2}{j}$
$L^\star$	$2^{m-f(A)-1} + 2^{m-2} - 1$	$2^{m-f(A)-2} + 2^{m-3}$
L^R	$\sum_{i=0}^{l-1} k^i + 2^{m-l} - 1$	$\sum_{i=0}^{l-1} k^i + 2^{m-l-1}$

Table 5: State complexity of basic regularity preserving operations on finite languages

Operation	itc
$L_1 \cup L_2$	$\sum_{\sigma\in\Sigma} (s_\sigma(L_1) \boxplus s_\sigma(L_2) - (\mathrm{itc}_\sigma(L_1) - s_\sigma(L_1))(\mathrm{itc}_\sigma(L_2) - s_\sigma(L_2)))\ +$ $n(\mathrm{itc}(L_1) - s(L_1)) + m(\mathrm{itc}(L_2) - s(L_2))$
$L_1 \cap L_2$	$\sum_{\sigma\in\Sigma} (s_\sigma(L_1)s_\sigma(L_2) + (\mathrm{itc}_\sigma(L_1) - s_\sigma(L_1)$ $- a_\sigma(L_1))(\mathrm{itc}_\sigma(L_2) - s_\sigma(L_2) - a_\sigma(L_2)) + a_\sigma(L_1)a_\sigma(L_2))$
L^C	$\lvert\Sigma\rvert(m+1)$
$L_1 L_2$	$k \sum_{i=0}^{m-2} \min\left\{k^i, \sum_{j=0}^{f(L_1,i)} \binom{n-1}{j}\right\} +$ $+ \sum_{\sigma\in\Sigma} \left(\min\left\{k^{m-1} - \overline{s}_\sigma(L_2), \sum_{j=0}^{f(L_1)-1} \Delta_j\right\} + \sum_{j=0}^{f(L_1)} \Lambda_j\right)$
$L^\star$	$2^{m-f(L)-1}\left(k + \sum_{\sigma\in\Sigma} 2^{e_\sigma(L)}\right) - \sum_{\sigma\in\Sigma} 2^{\overline{t}_\sigma(L) - \overline{s}_\sigma(L) - \overline{e}_\sigma(L)}$ $- \sum_{\sigma\in X} 2^{\overline{t}_\sigma(L) - \overline{s}_\sigma(L) - \overline{e}_\sigma(L)}$
L^R	$\sum_{i=0}^{l} k^i - 1 + k2^{m-l} - \sum_{\sigma\in\Sigma} 2^{\sum_{i=0}^{l-1} \overline{t}_\sigma(L,i)+1}$, m even
	$\sum_{i=0}^{l} k^i - 1 + k2^{m-l} - \sum_{\sigma\in\Sigma} \left(2^{\sum_{i=0}^{l-2} \overline{t}_\sigma(L,i)+1} - c_\sigma(l)\right)$, m odd

Table 6: Transition complexity of basic regularity preserving operations on finite languages

The level of A is the level of its pre-dead state. A DFA is called *linear* if its level is $n-1$.

4.1. Union

Consider the algorithm for the union operation based on the usual product construction already defined in the Section 3.1. Let $t_\sigma([k,l]) = \sum_{i\in[k,l]} t_\sigma(i)$. The following theorem presents the upper bounds for the number of states and transitions of any DFA accepting the union of two finite languages. Note that the result for the number of states is similar to the one for the complete case, omitting the dead state.

Theorem 10. *For any two finite languages L_1 and L_2 with $\mathrm{isc}(L_1) = m$ and $\mathrm{isc}(L_2) = n$, one has $\mathrm{isc}(L_1 \cup L_2) \leq mn - 2$ and*

$$\begin{aligned}\mathrm{itc}(L_1 \cup L_2) \leq \sum_{\sigma\in\Sigma} &(s_\sigma(L_1) \boxplus s_\sigma(L_2) - (\mathrm{itc}_\sigma(L_1) - s_\sigma(L_1))(\mathrm{itc}_\sigma(L_2) - s_\sigma(L_2)))\\ &+ n(\mathrm{itc}(L_1) - s(L_1)) + m(\mathrm{itc}(L_2) - s(L_2)),\end{aligned}$$

where for x, y Boolean values, $x \boxplus y = \min(x + y, 1)$.

Proof. Let $A = ([0, m[, \Sigma, \delta_A, 0, F_A)$ and $B = ([0, n[, \Sigma, \delta_B, 0, F_B)$ be the minimal DFAs that recognize L_1 and L_2, respectively. Let us consider, first, the counting of the number of states. In the product automaton, the set of states is a subset of $([0, m[\cup \{\Omega_A\}) \times ([0, n[\cup \{\Omega_B\})$. The states of the form $(0, i)$, where $i \in [1, n[\cup \{\Omega_B\}$, and of the form $(j, 0)$, where $j \in [1, m[\cup \{\Omega_A\}$, are not reachable from $(0, 0)$ because the operands represent finite languages; the states $(m-1, n-1)$, $(m-1, \Omega_B)$ and $(\Omega_A, n-1)$ are equivalent because they are final and they do not have out-transitions; the state (Ω_A, Ω_B) is the dead state and because we are dealing with incomplete DFAs we can ignore it. Therefore the number of states of the union of two incomplete DFAs accepting finite languages is at most $(m+1)(n+1) - (m+n) - 2 - 1 = mn - 2$.

Consider the number of transitions. In the product automaton, the σ-transitions can be represented as pairs (α_i, β_j) where α_i (respectively β_j) is 0 if there exists a σ-transition leaving the state i (respectively j) of DFA A (respectively B), or -1 otherwise. The resulting DFA can have neither transitions of the form $(-1, -1)$, nor of the form (α_0, β_j), where $j \in [1, n[\cup \{\Omega_B\}$, nor of the form (α_i, β_0), where $i \in [1, m[\cup \{\Omega_A\}$, as happened in the case of states. Thus, the number of σ-transitions for $\sigma \in \Sigma$ are:

$$\begin{aligned}s_\sigma(A) \boxplus s_\sigma(B) + &t_\sigma(A, [1, m[)t_\sigma(B, [1, n[) + t_\sigma(A, [1, m[)(\bar{t}_\sigma(B, [1, n[) + 1)\\ &+ (\bar{t}_\sigma(A, [1, m[) + 1)t_\sigma(B, [1, n[) =\\ s_\sigma(A) \boxplus s_\sigma(B) + &nt_\sigma(A, [1, m[) + mt_\sigma(B, [1, n[) - t_\sigma(A, [1, m[)t_\sigma(B, [1, n[).\end{aligned}$$

Because the DFAs are minimal, $\sum_{\sigma\in\Sigma} t_\sigma(A, [1, m[)$ corresponds to $\mathrm{itc}(L_1) - s(L_1)$, and analogously for B. Therefore the theorem holds. □

4.1.1. Worst-case Witnesses

In the following we show that the upper bounds described above are tight. Han and Salomaa proved [9, Lemma 3] that the upper bound for the number of states can not be reached for any alphabet with a fixed size. The witness families for the incomplete complexities coincide with the ones that these authors presented for the state complexity. As we are not including the dead state, our representation is slightly different. Let $m, n \geq 1$ and $\Sigma = \{b, c\} \cup \{a_{ij} \mid i \in [1, m[, j \in [1, n[, \ (i, j) \neq (m-1, n-1)\}$. Let $A = ([0, m[, \Sigma, \delta_A, 0, \{m-1\})$ where $\delta_A(i, b) = i+1$ for $i \in [0, m-2]$ and $\delta_A(0, a_{ij}) = i$ for $j \in [1, n[, \ (i, j) \neq (m-1, n-1)$. Let $B = ([0, n[, \Sigma, \delta_B, 0, \{n-1\})$, where $\delta_B(i, c) = i+1$ for $i \in [0, n[$ and $\delta_B(0, a_{i,j}) = j$ for $j \in [1, n[, i \in [1, m[, \ (i, j) \neq (m-1, n-1)$. See Figure 7 for the case $m = 5$ and $n = 4$.

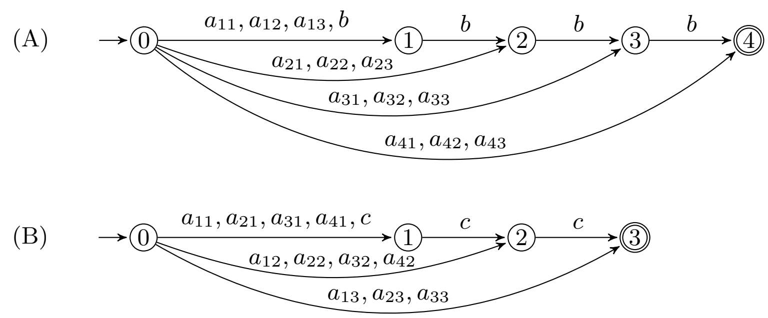

Figure 7: DFA A with $m = 5$ and DFA B with $n = 4$.

Theorem 11. *For any two integers $m \geq 2$ and $n \geq 2$, there exist an m-state DFA A and an n-state DFA B, both accepting finite languages, such that any DFA accepting $\mathcal{L}(A) \cup \mathcal{L}(B)$ needs at least $mn - 2$ states and $3(mn - n - m) + 2$ transitions, with an alphabet of size depending on m and n.*

Proof. The proof for the number of states is the same as the proof of [9, Lemma 2], considering the language families above. Let us prove the result for the number of transitions. The DFA A has $m-1$ b-transitions and one a_{ij}-transition, for each a_{ij}. The DFA B has $n-1$ c-transitions and the same number of a_{ij}-transitions as A. Thus, the DFA resulting for the union operation has:

- $mn - 2n + 1$ b-transitions;
- $mn - 2n + 1$ c-transitions;
- one a_{ij}-transitions for each a_{ij} and there are $mn - n - m$ different a_{ij}.

Thus, the total number of transitions is $3(mn - n - m) + 2$. It is easy to prove that the resulting DFA is minimal. □

4.2. Intersection

Given two DFAs $A = ([0, m[, \Sigma, \delta_A, 0, F_A)$ and $B = ([0, n[, \Sigma, \delta_B, 0, F_B)$, a DFA accepting $\mathcal{L}(A) \cap \mathcal{L}(B)$ can be also obtained by the product construction. Once more, the result for the state complexity is similar to the one for the complete case, omitting the dead state. Let $a_\sigma(A) = \sum_{i \in F} \mathrm{in}_\sigma(A, i)$, and $a(L) = \sum_{\sigma \in \Sigma} a_\sigma(L)$.

Theorem 12. *For any two finite languages L_1 and L_2 with $\mathrm{isc}(L_1) = m$ and $\mathrm{isc}(L_2) = n$, one has $\mathrm{isc}(L_1 \cap L_2) \leq mn - 2m - 2n + 6$ and*

$$\mathrm{itc}(L_1 \cap L_2) \leq \sum_{\sigma \in \Sigma} (s_\sigma(L_1)s_\sigma(L_2) + (\mathrm{itc}_\sigma(L_1) - s_\sigma(L_1) - a_\sigma(L_1))(\mathrm{itc}_\sigma(L_2) - s_\sigma(L_2) - a_\sigma(L_2)) + a_\sigma(L_1)a_\sigma(L_2))\,.$$

Proof. Let A and B be the minimal DFAs that recognize L_1 and L_2, respectively. Consider the DFA accepting $\mathcal{L}(A) \cap \mathcal{L}(B)$ obtained by the product construction. Let us prove the result for $\mathrm{isc}(L_1 \cap L_2)$. For the same reasons as in Theorem 10, we can eliminate the states of the form $(0, j)$, where $j \in [1, n[\cup \{\Omega_B\}$, and of the form $(i, 0)$, where $i \in [1, m[\cup \{\Omega_A\}$; the states of the form $(m-1, j)$, where $j \in [1, n-2]$, and of the form $(i, n-1)$, where $i \in [1, m-2]$ are equivalent to the state $(m-1, n-1)$ or to the state (Ω_A, Ω_B); the states of the form (Ω_A, j), where $j \in [1, n[\cup \{\Omega_B\}$, and of the form (i, Ω_B), where $i \in [1, m[\cup \{\Omega_A\}$ are equivalent to the state (Ω_A, Ω_B) which is the dead state of the DFA resulting from the intersection, and thus can be removed. Therefore, the number of states is at most $(m+1)(n+1) - 3((m+1)(n+1)) + 12 - 1 = mn - 2(m+n) + 6$.

Let us consider the $\mathrm{itc}(L_1 \cap L_2)$. Using the same technique as in Theorem 10 and considering that in the intersection we only have pairs of transitions where both elements are different from -1, the number of σ-transitions is as follows, which proves the theorem,

$$s_\sigma(A)s_\sigma(B) + (t_\sigma(A, [1, m[\,) \setminus \mathrm{in}_\sigma(A, F_A))(t_\sigma(B, [1, n[\,) \setminus \mathrm{in}_\sigma(B, F_B)) + a_\sigma(A)a_\sigma(B).$$

□

4.2.1. Worst-case Witnesses

The next result shows that the complexity upper bounds found above are reachable. The witness languages for the tightness of the bounds for this operation are different from the families given by Han and Salomaa, because those families are not tight for the transition complexity. For $m \geq 2$ and $n \geq 2$, let $\Sigma = \{a_{ij} \mid i \in [1, m-2],\ j \in [1, n-2]\} \cup \{a_{ij} \mid i = m-1,\ j = n-1\}$. Let $A = ([0, m[, \Sigma, \delta_A, 0, \{m-1\})$ where $\delta_A(x, a_{ij}) = x + i$ for $x \in [\,0, m[,\ i \in [1, m-2]$, and $j \in [1, n-2]$, and let $B = ([0, n[, \Sigma, \delta_B, 0, \{n-1\})$ where $\delta_B(x, a_{ij}) = x + j$ for $x \in [0, n[,\ i \in [1, m-2]$, and $j \in [1, n-2]$. The new families are presented in Figure 8 for $m = 5$ and $n = 4$.

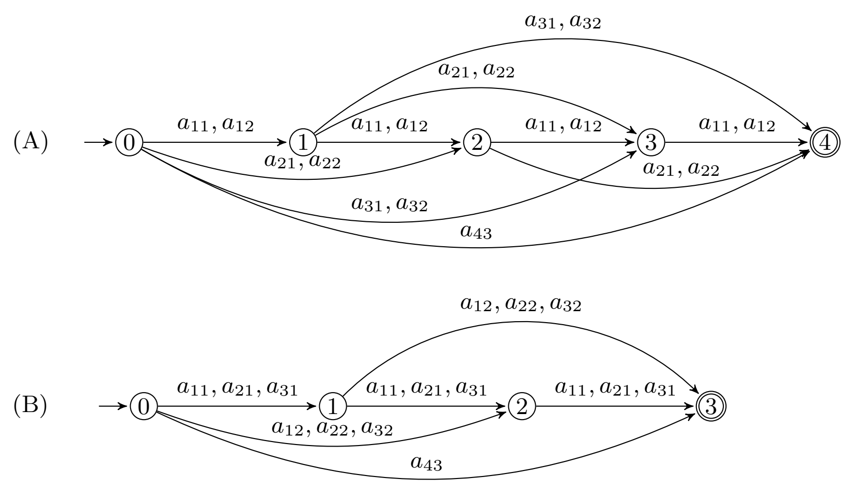

Figure 8: DFA A with $m = 5$ and DFA B with $n = 4$.

Theorem 13. *For any two integers $m \geq 2$ and $n \geq 2$, there exist an m-state DFA A and an n-state DFA B, both accepting finite languages, such that any DFA accepting $\mathcal{L}(A) \cap \mathcal{L}(B)$ needs at least $mn - 2(m+n) + 6$ states and $(m-2)(n-2)(2 + \sum_{i=1}^{\min(m,n)-3}(m-2-i)(n-2-i)) + 2$ transitions, with an alphabet of size depending on m and n.*

Proof. To prove that the minimal DFA accepting $L(A) \cap L(B)$ needs $mn - 2m - 2n + 6$ states we can use the same technique which is used in the proof of [9, Lemma 6]. For that, we define a set R of words which are not equivalent under $\equiv_{L(A)\cap L(B)}$. Let ε be the null string. We choose $R = R_1 \cup R_2 \cup R_3$, where $R_1 = \{\varepsilon\}$, $R_2 = \{a_{ij} \mid i = m-1,\ j = n-1\}$, and $R_3 = \{a_{ij} \mid i \in [1, m-2]$ and $j \in [1, n-2]\}$. It is easy to see that all words of each set are not equivalent to each other. As $|R_1| = |R_2| = 1$ and $|R_3| = (m-2)(n-2)$, we have that $|R| = mn - 2m - 2n + 6$. Thus the result for the number of states holds.

Let us consider the number of transitions. The DFA A has $(n-2)\sum_{i=0}^{m-3}(m-1-i) + 1$ a_{ij}- transitions. The DFA B has $(m-2)\sum_{i=0}^{n-3}(n-1-i) + 1$ a_{ij}-transitions. Let $k = (m-2)(n-2) + 1$. As in proof of Theorem 12, the DFA resulting from the intersection operation has the following number of transitions:

- k, corresponding to the pairs of transitions leaving the initial states of the operands;
- $\sum_{i=1}^{\min(m,n)-3}(n-2)(m-2-i)(m-2)(n-2-i)$, corresponding to the pairs of transitions formed by transitions leaving non-final and non-initial states of the operands;
- k, corresponding to the pairs of transitions leaving the final states of the operands.

Thus the total number of transitions is $2k + (m-2)(n-2)\sum_{i=1}^{\min(m,n)-3}(m-2-i)(n-2-i)$. $\square$

4.3. Complement

The state and transition complexity for this operation on finite languages are similar to the ones on regular languages [8]. This happens because the DFA must be completed. Let $A = ([0, m[, \Sigma, \delta_A, 0, F_A)$ be a DFA accepting the language L. The complement of L, L^c, is recognized by the DFA $C = ([0, m[\cup\{\Omega_A\}, \Sigma, \delta_C, 0, ([0, m[\setminus F_A) \cup \{\Omega_A\})$, where for $\sigma \in \Sigma$ and $i \in [0, m[$, $\delta_C(i, \sigma) = \delta_A(i, \sigma)$ if $\delta_A(i, \sigma)\uparrow$; $\delta_A(i, \sigma) = \Omega_A$ otherwise. Therefore one has,

Theorem 14. *For any finite language L with* $\mathrm{isc}(L) = m$ *one has* $\mathrm{isc}(L^C) \leq m + 1$ *and* $\mathrm{itc}(L^C) \leq |\Sigma|(m + 1)$.

Proof. Concerning the $\mathrm{isc}(L^C)$, it is only necessary to add a dead state to the operand DFA. The maximal number of σ-transitions is $m + 1$, because this is the number of states. Thus, the maximal number of transitions is $|\Sigma|(m + 1)$. $\square$

Gao et al. [8] gave the value $|\Sigma|(\mathrm{itc}(L) + 2)$ for the transition complexity of the complement. In some situations, this bound is higher than the bound here presented, but contrasting to that one, it gives the transition complexity of the operation as a function of the transition complexity of the operand.

The witness family for this operation is exactly the same presented in [8], i.e. $\{b^m\}$, for $m \geq 1$. It is easy to see that the bounds are tight for this family.

4.4. Concatenation

Câmpeanu *et al.* [5] studied the state complexity of the concatenation of an m-state complete DFA A with an n-state complete DFA B over an alphabet of size k and proposed the upper bound

$$\sum_{i=0}^{m-2} \min\left\{k^i, \sum_{j=0}^{f(A,i)} \binom{n-2}{j}\right\} + \min\left\{k^{m-1}, \sum_{j=0}^{f(A)} \binom{n-2}{j}\right\}, \tag{4}$$

where $f(A, i)$ is the larger number of final states of any path from the initial state to the state i. They proved that this upper bound is tight for $m > n - 1$. It is easy to see that the second term of (4) is $\sum_{j=0}^{f(A)} \binom{n-2}{j}$ if $m > n - 1$, and k^{m-1}, otherwise. The value k^{m-1} indicates that the DFA resulting from the concatenation has states with level at most $m - 1$. But that is not always the case, as we can see by the example[4] in Figure 9. This implies that (4) is not an upper bound if $m < n$. Thus, we have

[4]Note that we are omitting the dead state in the figures.

Theorem 15. *For any two finite languages L_1 and L_2 with $\mathrm{sc}(L_1) = m$ and $\mathrm{sc}(L_2) = n$ over an alphabet of size $k \geq 2$, one has*

$$\mathrm{sc}(L_1L_2) \leq \sum_{i=0}^{m-2} \min \left\{ k^i, \sum_{j=0}^{f(L_1,i)} \binom{n-2}{j} \right\} + \sum_{j=0}^{f(L_1)} \binom{n-2}{j}. \tag{5}$$

Proof. The proof follows the one in [5] considering the changes described above. □

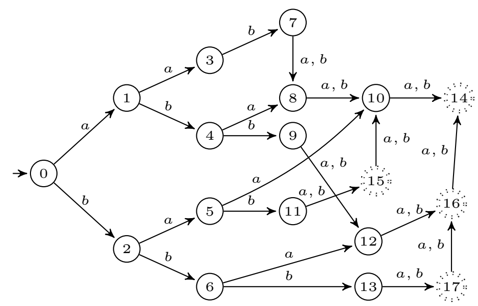

Figure 9: DFA resulting from the concatenation of DFA A with $m = 3$ and DFA B with $n = 5$, of Fig. 11. The states with dashed lines have level > 3 and are not accounted by formula (4).

Consider the algorithm for the concatenation presented in the Section 3.2, and let $\overline{s}_\sigma(A) = \overline{t}_\sigma(A, 0)$. The next theorem presents the upper bounds for the number of states and transitions of any DFA accepting L_1L_2. Note that the result for the number of states is similar to the Theorem 15, omitting the dead state.

Theorem 16. *For any two finite languages L_1 and L_2 with $\mathrm{isc}(L_1) = m$ and $\mathrm{isc}(L_2) = n$ over an alphabet of size $k \geq 2$, and making $\Lambda_j = \binom{n-1}{j} - \binom{\overline{t}_\sigma(L_2) - \overline{s}_\sigma(L_2)}{j}$, $\Delta_j = \binom{n-1}{j} - \left(\binom{\overline{t}_\sigma(L_2) - \overline{s}_\sigma(L_2)}{j} * \overline{s}_\sigma(L_2)\right)$ one has*

$$\mathrm{isc}(L_1L_2) \leq \sum_{i=0}^{m-1} \min \left\{ k^i, \sum_{j=0}^{f(L_1,i)} \binom{n-1}{j} \right\} + \sum_{j=0}^{f(L_1)} \binom{n-1}{j} - 1 \tag{6}$$

and

$$\mathrm{itc}(L_1L_2) \leq k \sum_{i=0}^{m-2} \min \left\{ k^i, \sum_{j=0}^{f(L_1,i)} \binom{n-1}{j} \right\} + \\ + \sum_{\sigma \in \Sigma} \left(\min \left\{ k^{m-1} - \overline{s}_\sigma(L_2), \sum_{j=0}^{f(L_1)-1} \Delta_j \right\} + \sum_{j=0}^{f(L_1)} \Lambda_j \right). \tag{7}$$

Proof. Let $A = ([0,m[,\Sigma,\delta_A,0,F_A)$ and $B = ([0,n[,\Sigma,\delta_B,0,F_B)$ be the minimal DFAs that recognize L_1 and L_2. Consider the DFA C accepting $\mathcal{L}(A)\mathcal{L}(B)$. Let us prove the result for $\mathrm{isc}(L_1L_2)$. Each state of the DFA C has the form (x,P) where $x \in [0,m[\cup\{\Omega_A\}$ and $P \subseteq [0,n[$. The first term of (6) corresponds to the maximal number of states of the form (i,P) with $i \in [0,m[$. Such a state (i,P) is at a level $\le i$, which has at most k^{i-1} predecessors. Thus, the level i has at most k^i states. The maximal size of the set P is $f(A,i)$. For a fixed i, the initial state of the DFA B either belongs to all sets P (if $i \in F_A$) or it is not in any of them. Thus, the number of distinct sets P is at most $\sum\limits_{j=0}^{f(A,i)} \binom{n-1}{j}$. The number of states of the form (i,P) is the minimal of these two values. The second term of (6) corresponds to the maximal number of states where the first component is Ω_A. In this case, the size of P is at most $f(A)$. Lastly, we remove the dead state.

Consider now the result for $\mathrm{itc}(L_1L_2)$. The σ-transitions of the DFA C have three forms: (i,β) where i represents the transition leaving the state $i \in [0,m[$; $(-1,\beta)$ where -1 represents the absence of the transition from state $m-1$ to Ω_A; and $(-2,\beta)$ where -2 represents any transition leaving Ω_A. In all forms, β is a set of transitions of DFA B. The number of σ-transitions of the form (i,β) is at most $\sum_{i=0}^{m-2}\min\{k^i,\sum_{j=0}^{f(L_1,i)}\binom{n-1}{j}\}$ which corresponds to the number of states of the form (i,P), for $i \in [0,m[$ and $P \subseteq [0,n[$. The number of σ-transitions of the form $(-1,\beta)$ is $\min\{k^{m-1}-\overline{s}_\sigma(L_2),\sum_{j=0}^{f(L_1)-1}\Delta_j\}$. We have at most k^{m-1} states in this level. However, if $s_\sigma(B,0)=0$ we need to remove the transition $(-1,\emptyset)$ which leaves the state $(m-1,\{0\})$. On the other hand, the size of β is at most $f(L_1)-1$ and we know that β has always the transition leaving the initial state by σ, if it exists. If this transition does not exist, *i.e.* $\overline{s}_\sigma(B,0)=1$, we need to remove the sets with only non-defined transitions, because they originate transitions of the form $(-1,\emptyset)$. The number of σ-transitions of the form $(-2,\beta)$ is $\sum_{j=0}^{f(L_1)}\Lambda_j$ and this case is similar to the previous one. $\square$

4.4.1. Worst-case Witnesses

To prove that the bounds are reachable, we consider two cases depending whether $m+1 \ge n$ or not.

Case 1: $m+1 \ge n$. The witness languages are the ones presented by Câmpeanu et al. (see Figure 10).

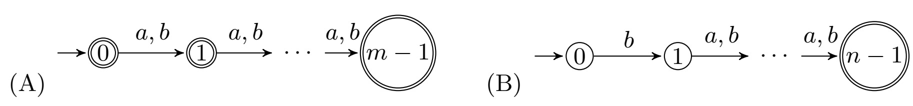

Figure 10: DFA A with m states and DFA B with n states.

Theorem 17. *For any two integers $m \ge 2$ and $n \ge 2$ such that $m+1 \ge n$, there exist an m-state DFA A and an n-state DFA B, both accepting finite languages,* *such that any DFA accepting $\mathcal{L}(A)\mathcal{L}(B)$ needs at least $(m-n+3)2^{n-1}-2$ states and $2^n(m-n+3)-8$ transitions.*

Proof. The proof for the number of states is similar to the proof of [5, Theorem 4]. Let us consider the number of transitions. The DFA A has $m-1$ σ-transitions for each $\sigma \in \{a,b\}$. The number of final states in the DFA A is m. The DFA B has $n-2$ a-transitions and $n-1$ b-transitions. Consider $m \geq n$. If we analyse the transitions as we did in the proof of the Theorem 16 we have $2^{n-1}(m-n+1)-1$ a-transitions and $2^{n-1}(m-n+1)-1$ b-transitions that correspond to the transitions of the form (i,β); $2^{n-1}-2$ a-transitions and 2^{n-1} b-transitions that correspond to the transitions of the form $(-1,\beta)$; and $2^{n-1}-2$ a-transitions and $2^{n-1}-2$ b-transitions that correspond to the transitions of the form $(-2,\beta)$. Thus, we calculate that the total number of transitions is

$$\begin{aligned}&2(2^{n-1}(m-n+1)-1)+2^{n-1}-2+2^{n-1}-2+2^{n-1}+2^{n-1}-2\\&=2^n(m-n+3)-8.\end{aligned}$$

□

Case 2: $m+1<n$. Let $\Sigma=\{b\}\cup\{a_i \mid i\in[1,n-2]\}$. Let $A=([0,m[,\Sigma,\delta_A,0,[0,m[)$ where $\delta_A(i,\sigma)=i+1$, for any $\sigma\in\Sigma$. Let $B=([0,n[,\Sigma,\delta_B,0,\{n-1\})$ where $\delta_B(i,b)=i+1$, for $i\in[0,n-2]$, $\delta_B(i,a_j)=i+j$, for $i,j\in[1,n-2]$, $i+j\in[2,n[$, and $\delta_B(0,a_j)=j$, for $j\in[2,n-2]$.

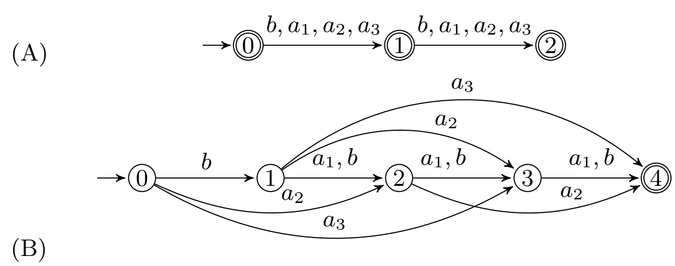

Figure 11: DFA A with $m=3$ states and DFA B with $n=5$ states.

Theorem 18. *For any two integers $m\geq 2$ and $n\geq 2$, with $m+1<n$, there exist an m-state DFA A and an n-state DFA B, both accepting finite languages over an alphabet of size depending on m and n, such that the number of states and transitions of any DFA accepting $\mathcal{L}(A)\mathcal{L}(B)$ reaches the upper bounds.*

Proof. We need to show that the DFA C, resulting from the concatenation algorithm already defined and accepting $\mathcal{L}(A)\mathcal{L}(B)$, is minimal, i.e. (i) every state of C is reachable from the initial state; (ii) each state of C defines a distinct equivalence class. To prove (i), we first show that all states $(i,P)\subseteq R$ with $i\in[1,m[$ are reachable. The following facts hold for the automaton C:

1. every state of the form $(i+1,P')$ is reached by a transition from a state (i,P) (by the construction of A) and $|P'|\leq|P|+1$, for $i\in[1,m-2]$;

2. every state of the form (Ω_A, P') is reached by a transition from a state $(m-1, P)$ (by the construction of A) and $|P'| \leq |P| + 1$;

3. for each state (i, P), $P \subseteq [0, n[$, $|P| \leq i + 1$ and $0 \in P$, $i \in [1, m[$;

4. for each state (Ω_A, P), $\emptyset \neq P \subseteq [0, n[$, $|P| \leq m$ and $0 \notin P$.

Suppose that for a $i \in [1, m-2]$, all states (i, P) are reachable. The number of states of the form $(1, P)$ is $m-1$ and of the form (i, P) with $i \in [2, m-2]$ is $\sum_{j=0}^{i} \binom{n-1}{j}$. Let us consider the states $(i+1, P')$. If $P' = \{0\}$, then $\delta_C((i, \{0\}), a_1) = (i+1, P')$. Otherwise, let $l = \min(P' \setminus \{0\})$ and $S_l = \{s - l \mid s \in P' \setminus \{0\}\}$. Then,

$$\begin{aligned} \delta_C((i, S_l), a_l) &= (i+1, P') & \text{if} \quad & 2 \leq l \leq n-2 \\ \delta_C((i, \{0\} \cup S_1), a_1) &= (i+1, P') & \text{if} \quad & l = n-1 \\ \delta_C((i, S_1), b) &= (i+1, P') & \text{if} \quad & l = 1 \end{aligned}$$

Thus, all $\sum_{j=0}^{i+1} \binom{n-1}{j}$ states of the form $(i+1, P')$ are reachable. Let us consider the states (Ω_A, P'). P' is always a non-empty set by construction of C. Let $l = \min(P')$ and $S_l = \{s - l \mid s \in P'\}$. Thus,

$$\begin{aligned} \delta_C((m-1, S_l), a_l) &= (\Omega_A, P') & \text{if} \quad & 2 \leq l \leq n-2 \\ \delta_C((m-1, \{0\} \cup S_1), a_1) &= (\Omega_A, P') & \text{if} \quad & l = n-1 \\ \delta_C((m-1, S_1), b) &= (\Omega_A, P') & \text{if} \quad & l = 1 \end{aligned}$$

Thus, all $\sum_{j=0}^{m} \binom{n-1}{j} - 1$ states of the form (Ω_A, P') are reachable.

To prove (ii), consider two distinct states $(i, P_1), (j, P_2) \in R$. If $i \neq j$, then $\delta_C((i, P_1), b^{n+m-2-i}) \in F_C$ but $\delta_C((j, P_2), b^{n+m-2-i}) \notin F_C$. If $i = j$, suppose that $P_1 \neq P_2$ and both are final or non-final. Let $P_1' = P_1 \setminus P_2$ and $P_2' = P_2 \setminus P_1$. Without loss of generality, let P_1' be the set which has the minimal value, let us say l. Thus $\delta_C((i, P_1), a_1^{n-1-l}) \in F_C$ but $\delta_C((i, P_2), a_1^{n-1-l}) \notin F_C$. Thus C is minimal.

Let us consider the number of transitions. The DFA A has $m-1$ σ-transitions, for $\sigma \in \Sigma$. The DFA B has $n-1$ b-transitions, $n-2$ a_1-transitions, and $n-i$ a_i-transitions, with $i \in [2, n-2]$. Thus DFA A has $|\Sigma|(m-1)$ transitions, DFA B has $2n - 3 + \sum_{i=2}^{n-2}(n-i)$ transitions and $|\Sigma| = n-1$. The proof is similar to the proof of Theorem 16. □

Theorem 19. *The upper bounds for state and transition complexity of concatenation presented in Theorem 16 cannot be reached for any alphabet with a fixed size for $m \geq 0$, $n > m+1$.*

Proof. Consider the construction for the concatenation presented in the Section 3.2. Let us define the subset $S = \{(\Omega_A, P) \mid 1 \in P\}$ of R. In order for a state (Ω_A, P) to belong to S it has to satisfy the following condition:

$$\exists i \in F_A \exists P' \subseteq 2^{[0,n[} \exists \sigma \in \Sigma : \delta_C((i, P' \cup \{0\}), \sigma) = (\Omega_A, P).$$

The maximal size of S is $\sum_{j=0}^{f(A)-1}\binom{n-2}{j}$, because by construction $1 \in P$ and $0 \notin P$. Assume that Σ has a fixed size $k = |\Sigma|$. Then, the maximal number of words that reach states of S from r_0 is $\sum_{i=0}^{f(A)} k^{i+1}$ since the words that reach a state $s \in S$ are of the form $w_A\sigma$, where $w_A \in L(A)$ and $\sigma \in \Sigma$. As $n > m$, for some $l \geq 0$ we have $n = m + l$. Thus for an l sufficiently large $\sum_{i=0}^{f(A)} k^{i+1} \ll \sum_{j=0}^{f(A)-1}\binom{m+l-2}{j}$, which is absurd and resulted from supposing that k is fixed. □

4.5. Kleene Star

Consider the algorithm for the Kleene star operation presented in the Section 3.3. If $f(A) = 1$ then $\mathcal{L}(A)^\star = \mathcal{L}(A)$. Thus, we will consider DFAs with at least two final states. Let $e_\sigma(A) = \sum_{i\in F} t_\sigma(A,i)$ and $\bar{e}_\sigma(A) = \sum_{i\in F} \bar{t}_\sigma(A,i)$. The following results give the number of states and transitions which are sufficient for any DFA B accepting $\mathcal{L}(A)^\star$.

Theorem 20. *For any finite language L with* $\mathrm{isc}(L) = m$ *and* $f(L) \geq 2$*, one has* $\mathrm{isc}(L^\star) \leq 2^{m-f(L)-1} + 2^{m-2} - 1$ *and*

$$\mathrm{itc}(L^\star) \leq 2^{m-f(L)-1}\left(k + \sum_{\sigma\in\Sigma} 2^{e_\sigma(L)}\right) - \sum_{\sigma\in\Sigma} 2^{n_\sigma} - \sum_{\sigma\in X} 2^{n_\sigma},$$

where $n_\sigma = \bar{t}_\sigma(L) - \bar{s}_\sigma(L) - \bar{e}_\sigma(L)$ and $X = \{\sigma \in \Sigma \mid s_\sigma(L) = 0\}$.

Proof. The proof for the states is similar to the proof presented by Câmpeanu et al. [5]. Let $A = ([0,m[, \Sigma, \delta_A, 0, F_A)$ be the minimal DFA that recognize L. Note that in the star operation the states of the resulting DFA are sets of states of the DFA A. The minimal DFA B accepting $L(A)^\star$, obtained by the referred algorithm, has at most the following states:

(i) the initial state 0_B which corresponds to the initial state of A: 1 state;

(ii) all $P \subseteq [1,m[\setminus F_A$ and $P \neq \emptyset$: $2^{m-f(A)-1} - 1$ states;

(iii) all $P \subseteq [0,m-2]$ such that $P \cap F_A \neq \emptyset$ and $0 \in P$: $2^{m-f(A)-1}(2^{f(A)-1}-1)$ states;

(iv) all $P = P' \cup \{m-1, 0\}$ where $P' \subseteq [1,m[\setminus F_A$ and $P' \neq \emptyset$: $2^{m-f(A)-1} - 1$ states.

Therefore, the number of states of the DFA B is at most $2^{m-f(A)-1} + 2^{m-2} - 1$. As in [5, Theorem 1], in the above description we are considering that $0 \notin F_A$. If $0 \in F_A$ the values suffer a few changes but the formula which is obtained, when reaches its maximum, is the same.

The proof for the $\mathrm{itc}(L^\star)$ is similar to the one for the $\mathrm{isc}(L^\star)$. Enumerating the σ-transitions as done for the states, we have that:

(i) the presence or the absence of the transition leaving the initial state: $s_\sigma(L)$ σ-transitions;

(ii) the set of transitions leaving non-initial and non-final states: $2^{m-f(L)-1} - 2^{\overline{t}_\sigma(L)-\overline{s}_\sigma(L)-\overline{e}_\sigma(L)}$;

(iii) the set of transitions leaving the final states (excluding the pre-dead): $2^{m-f(L)-1}(2^{e_\sigma(L)} - 1)$ σ-transitions;

(iv) the set of transitions leaving the pre-dead state: $2^{m-f(L)-1}-1$ σ-transitions if there exists a σ-transition leaving the initial state, $2^{m-f(L)-1} - 2^{n_\sigma}$ σ-transitions otherwise, where $n_\sigma = \overline{t}_\sigma(L) - \overline{s}_\sigma(L) - \overline{e}_\sigma(L)$.

Thus the upper bound for $\text{itc}(L^\star)$ holds. $\square$

4.5.1. Worst-case Witnesses

The theorem below shows that the previous upper bounds are reachable. The witness family for this operation is the same as the one presented by Câmpeanu et al., but we have to exclude the dead state.

Let $A = ([0, m[, \{a, b, c\}, \delta_A, 0, \{m-2, m-1\})$, $m \geq 4$, be a incomplete DFA accepting a finite language (see Figure 12) where:

$$\begin{aligned}
\delta(i, a) &= i + 1, \text{ for } i \in [0, m[\\
\delta(i, b) &= i + 1, \text{ for } i \in [1, m[\text{ and } \delta(0, b) = m - 1 \\
\delta(i, c) &= i + 1, \text{ for } i \in [0, m[\text{ and } m - i \text{ is even.}
\end{aligned}$$

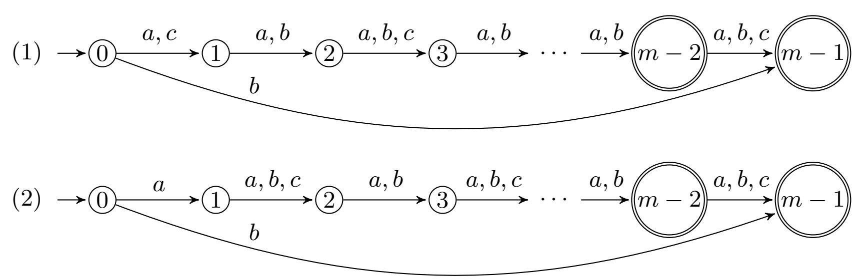

Figure 12: DFA A with m states, with m even (1) and odd (2).

Theorem 21. *For any integer $m \geq 4$, there exist an m-state DFA A accepting a finite language, such that any DFA accepting $\mathcal{L}(A)^\star$ needs at least $2^{m-2}+2^{m-3}-1$ states and $9\cdot2^{m-3}-2^{m/2}-2$ transitions if m is odd, or $9\cdot2^{m-3}-2^{(m-2)/2}-2$ transitions, otherwise.*

Proof. The proof for the states is the same as presented by Câmpeanu et al.. Note that we do not count the dead states, and because of this we have one state less in A and in the resulting DFA. Considering the transitions as in the proof of Theorem 20, the DFA resulting for the star operation has: $3 \cdot 2^{m-3} - 1$ a-transitions, $3 \cdot 2^{m-3} - 1$ b-transitions, and $3 \cdot 2^{m-3} - 2^{m/2}$ c-transitions if m

is odd, or $3 \cdot 2^{m-3} - 2^{(m-2)/2}$ transitions otherwise. Therefore the resulting DFA has $9 \cdot 2^{m-3} - 2^{m/2} - 2$ transitions if m is odd, or $9 \cdot 2^{m-3} - 2^{(m-2)/2} - 2$ transitions, otherwise. $\square$

4.6. Reversal

Given an incomplete DFA $A = ([0, m[, \Sigma, \delta_A, 0, F_A)$, to obtain a DFA B that accepts $\mathcal{L}(A)^R$, we first reverse all transitions of A and then determinize the resulting NFA. Let $c_\sigma(A, i) = 0$ if $\mathrm{in}_\sigma(A, i) > 0$ and 1 otherwise. In the following result we present upper bounds for the number of states and transitions of B.

Theorem 22. *For any finite languages L with $\mathrm{isc}(L) = m$, $m \geq 3$, and over an alphabet of size $k \geq 2$, , where l is the smallest integer such that $2^{m-l} \leq k^l$, one has $\mathrm{isc}(L^R) \leq \sum_{i=0}^{l-1} k^i + 2^{m-l} - 1$ and if m is odd,*

$$\mathrm{itc}(L^R) \leq \sum_{i=0}^{l} k^i - 1 + k2^{m-l} - \sum_{\sigma \in \Sigma} 2^{\sum_{i=0}^{l-1} \bar{t}_\sigma(L,i)+1},$$

or, if m is even,

$$\mathrm{itc}(L^R) \leq \sum_{i=0}^{l} k^i - 1 + k2^{m-l} - \sum_{\sigma \in \Sigma} \left(2^{\sum_{i=0}^{l-2} \bar{t}_\sigma(L,i)+1} - c_\sigma(L, l)\right).$$

Proof. Let A be the minimal DFA accepting L. The proof for $\mathrm{isc}(L^R)$ is similar to the proof of [5, Theorem 5]. We only need to remove the dead state.

Let us prove the result for $\mathrm{itc}(L^R)$. The smallest l that satisfies $2^{m-l} \leq k^l$ is the same for m and $m+1$, and because of that we have to consider whether m is even or odd.

Suppose m odd. Let T_1 be the set of transitions corresponding to the first $\sum_{i=0}^{l-1} k^i$ states and T_2 be the set corresponding to the other $2^{m-l} - 1$ states. We have that $|T_1| = \sum_{i=0}^{l-1} k^i - 1$, because the initial state has no transition reaching it. As the states of DFA B are sets of states of DFA A, we also consider each σ-transition of B as a set of σ-transitions of A. If all σ-transitions were defined in A, T_2 would have 2^{m-l} σ-transitions. But, as not all σ-transitions are defined, we remove from 2^{m-l} the sets which only have undefined σ-transitions of A. As the initial state of A has no transitions reaching it, we need to add one to the number of undefined σ-transitions. Thus, $|T_2| = \sum_{\sigma \in \Sigma} 2^{m-l} - 2^{(\sum_{i=0}^{l-1}(\bar{t}_\sigma(i)))+1}$.

Let us consider m even. In this case we also need to consider the set of transitions that connect the states with the highest level in the first set (T_1) with the states with the lowest level in the second set (T_2). As the highest level is $l-1$, we have to remove the possible transitions that reach the state l in DFA A. $\square$

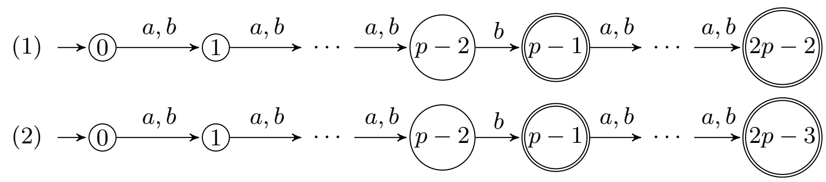

Figure 13: DFA A with $m = 2p - 1$ states (1) and with $m = 2p - 2$ (2).

4.6.1. Worst-case Witnesses

The following result proves that the upper bounds presented above are tight. The witness family for this operation is the one presented by Câmpeanu et al. but we omit the dead state. It is depicted in Figure 13.

Theorem 23. *For any integer $m \geq 4$, there exist an m-state DFA A accepting a finite language, such that any DFA accepting $\mathcal{L}(A)^R$ needs at least $3 \cdot 2^{p-1} + 2$ states and $3 \cdot 2^p - 8$ transitions if $m = 2p - 1$ or $2^{p+1} - 2$ states and $2^{p+2} - 7$ transitions if $m = 2p$.*

Proof. The proof for the states is the same as the one presented by Câmpeanu et al. [5]. Considering the transitions as in the proof of Theorem 22, the DFA resulting for the reversal operation, in case $m = 2p - 1$, has:

- $(\sum_{i=0}^{p-1} 2^i) - 1$ transitions in T_1;
- $2^p - 2^2$ a-transitions in T_2;
- $2^p - 2$ b-transitions in T_2.

Thus, the resulting DFA has $3 \cdot 2^p - 8$ transitions. In the other case, the resulting DFA has:

- $(\sum_{i=0}^{p-1} 2^i) - 1$ transitions in T_1;
- $2^p - 2$ a-transitions in T_2;
- $2^{p-1} - 1$ a-transitions in the intermediate set;
- $2^p - 2$ b-transitions in T_2;
- 2^{p-1} b-transitions in the intermediate set.

Therefore the resulting DFA has $2^{p+2} - 7$ transitions. □

4.7. Experimental Results

Similarly to the previous section, we performed some experimental tests in order to analyse the practical behaviour of the operations over finite languages. All the tests were performed with uniformly random generated acyclic DFAs.

Table 7 shows the results of 20000 experimental tests. The number of states of the operands and the measures are the same as used in Section 3.6.

Concatenation												
m	n		ubsc	rs	m_1	m_2	tc	ubtc	rt	m_3	m_4	d
2	18	37.11	88.64	0.42	108	159	163.43	417.57	0.39	530	786	0.85
4	16	63.26	634.06	0.1	236	2096	289.27	2913.99	0.10	1147	10471	0.89
6	14	76.09	2480.07	0.03	268	7256	350.93	8249.58	0.04	1305	36267	0.91
8	12	78.28	3803.77	0.02	252	7024	360.60	11050.80	0.03	1236	35105	0.91
10	10	73.52	2670.73	0.03	260	3296	336.23	8314.19	0.04	1285	16463	0.91
12	8	63.8	1158.59	0.06	170	1208	287.97	4143.13	0.07	837	6031	0.90
14	6	51.2	396.78	0.13	123	398	226.61	1615.26	0.14	600	1981	0.88
16	4	37.69	122.99	0.31	75	123	162.18	540.69	0.30	363	610	0.86
18	2	25.09	36	0.70	33	36	104.01	165.38	0.63	152	175	0.83
Union												
10	10	30.95	98	0.32	57	98	125.41	8314.19	0.02	260	16463	0.81
12	8	29.86	94	0.32	52	94	120.75	416.94	0.29	225	455	0.81
14	6	26.55	82	0.32	47	82	106.98	360.17	0.30	203	395	0.80
16	4	21.84	62	0.35	36	62	88.5	267.03	0.33	151	297	0.81
18	2	18.8	34	0.55	22	34	77.06	142.41	0.54	97	163	0.82
Intersection												
10	10	12.93	66	0.20	33	66	29.54	110.7	0.27	106	256	0.44
12	8	11.91	62	0.19	33	62	26.71	102.39	0.26	92	239	0.43
14	6	9.02	50	0.18	25	50	18.96	79.85	0.24	79	168	0.40
16	4	5.02	30	0.17	14	30	8.76	43.92	0.20	39	114	0.33
18	2	1.78	2	0.89	2	2	1.29	2.47	0.52	5	5	0.13
Star												
2		1	0.75	1.33	1	1	2.59	1.94	1.33	5	5	0.52
4		3.05	4.67	0.65	5	7	11.81	15.07	0.78	25	35	0.78
6		7.43	18.82	0.40	23	31	32.86	65.17	0.50	112	154	0.88
8		14.59	71.46	0.20	73	127	68.05	241.79	0.28	362	631	0.93
10		25.11	274.14	0.092	121	511	120.19	888.33	0.135	598	2549	0.955
12		38.75	1066.12	0.036	192	2047	188.05	3297.08	0.057	949	10226	0.969
14		57.18	4190.58	0.014	416	8191	279.82	12436.48	0.023	2078	40896	0.977
16		79.35	16599.54	0.005	481	32767	390.42	47644.04	0.008	2400	163810	0.982
18		108.37	66019.6	0.002	751	98303	535.28	184747.27	0.003	3745	491492	0.986
Reversal												
2		2	2	1	2	2	2.59	2.59	1	5	5	0.26
4		5.58	7.99	0.70	8	8	12.93	31.72	0.41	29	35	0.46
6		11.87	20.10	0.57	20	21	33.96	96.08	0.35	76	100	0.57
8		21.99	62.00	0.35	44	62	70.66	298.14	0.24	182	305	0.63
10		37.35	158	0.24	94	158	129.88	779.05	0.17	401	785	0.69
12		59.34	411	0.14	144	411	217.31	2042.01	0.11	640	2050	0.72
14		89.91	1179	0.08	247	1179	342.91	5882.71	0.06	1115	5890	0.75
16		130.19	2828	0.05	355	2828	511.46	14126.25	0.04	1629	14135	0.78
18		184.32	8001	0.02	460	8001	742.56	39989.92	0.02	2057	40000	0.80
Complement												
2		3	3	1	3	3	7.77	8	1	8	8	1
4		5	5	1	5	5	24.12	25	1	25	25	1
6		7	7	1	7	7	34.97	35	1	35	35	1
8		9	9	1	9	9	45	45	1	45	45	1
10		11	11	1	11	11	55	55	1	55	55	1
12		13	13	1	13	13	65	65	1	65	$6s5$	1
14		15	15	1	15	15	75	75	1	75	75	1
16		17	17	1	17	17	85	85	1	85	85	1
18		19	19	1	19	19	95	95	1	95	95	1

Table 7: Experimental results for finite languages.

The results obtained were similar to the ones for regular languages. However, for finite languages, the difference between the worst and the average case was not as high as for regular languages. For example, for reversal operation, considering $m = 18$ and regular languages, the upper bound for the number of states was 3700 times larger than the number of states observed and the upper bound for the number of transitions was 5600 times larger than the number of transitions, whereas for finite languages the upper bound for states was only 43 times larger and for transitions 53 times larger. As for regular languages, the DFAs resulting from all the operations (excluding the complement) were also incomplete.

Thus, as what happened for regular languages, we can conjecture that the upper bounds are seldom reached in practical applications.

5. Final Remarks

In this paper we presented tight upper bounds for the incomplete state and transition complexities for union, concatenation, Kleene star, complement and reversal on general and finite regular languages. Transition complexity bounds are expressed as functions of several more fine-grained measures of the operands, such as the number of final states, the number of undefined transitions or the number of transitions that leave the initial state. Table 1 summarizes the results for incomplete transition complexity, using the witnesses parameters.

Tables 2 and 3 summarize some of the results on state complexity and transition complexity of basic operations on general regular languages, respectively. In Table 2 we present the state complexity (sc), based on complete DFA [26], incomplete DFA (isc), the new results here presented, and the ones from Gao et al. [8]; finally, the results for state complexity for NFAs (nsc) [10]. The upper bound for the nondeterministic transition complexity of the complement is not tight, and thus we inscribe the corresponding lower and the upper bounds.

Table 5 and Table 6 have the formulae for the upper bounds of state and transition complexity for all the studied operations on finite regular languages.

The experimental results for both cases show that the upper bounds for state and transition complexities are much higher than the observed number of states and transitions of the DFAs resulting from the operations, with uniform random generated operands. Thus, although the study of the descriptional complexities considering the worst-case analysis is fundamental, in order to have good estimates of the amount of resources required to manipulate representations of a given language in practical applications, average-case complexity results need to be considered.

References

[1] Almeida, M., Moreira, N., Reis, R., 2007. Enumeration and generation with a string automata representation. Theor. Comput. Sci. 387 (2), 93–102.

[2] Beesley, K. R., Karttunen, L., 2003. Finite State Morphology. CSLI Publications, Stanford University.

[3] Bordihn, H., Holzer, M., Kutrib, M., 2009. Determination of finite automata accepting subregular languages. Theor. Comput. Sci. 410 (35), 3209–3222.

[4] Brzozowski, J. A., 2010. Complexity in convex languages. In: Dediu, A. H., Fernau, H., Martín-Vide, C. (Eds.), Proc. 4th LATA. Vol. 6031 of LNCS. Springer, pp. 1–15.

[5] Câmpeanu, C., Culik II, K., Salomaa, K., Yu, S., 2001. State complexity of basic operations on finite languages. In: Boldt, O., Jürgensen, H. (Eds.), Proc. 4th WIA. Vol. 2214 of LNCS. Springer, pp. 60–70.

[6] Cassandras, C. G., Lafortune, S., 2006. Introduction to discrete event systems. Springer.

[7] Domaratzki, M., Salomaa, K., 2007. Transition complexity of language operations. Theor. Comput. Sci. 387 (2), 147–154.

[8] Gao, Y., Salomaa, K., Yu, S., 2011. Transition complexity of incomplete DFAs. Fundam. Inform. 110 (1-4), 143–158.

[9] Han, Y., Salomaa, K., 2008. State complexity of union and intersection of finite languages. Int. J. Found. Comput. Sci. 19 (3), 581–595.

[10] Holzer, M., Kutrib, M., 2003. State complexity of basic operations on nondeterministic finite automata. In: Champarnaud, J.-M., Maurel, D. (Eds.), Proc. 7th CIAA. Vol. 2608 of LNCS. Springer, pp. 148–157.

[11] Holzer, M., Kutrib, M., 2009. Descriptional and computational complexity of finite automata. In: Dediu, A. H., Ionescu, A.-M., Martín-Vide, C. (Eds.), Proc. 3rd LATA. Vol. 5457 of LNCS. Springer, pp. 23–42.

[12] Holzer, M., Kutrib, M., 2009. Nondeterministic finite automata - recent results on the descriptional and computational complexity. Int. J. Found. Comput. Sci. 20 (4), 563–580.

[13] Hopcroft, J. E., Ullman, J. D., 1979. Introduction to Automata Theory, Languages and Computation. Addison-Wesley.

[14] Jirásková, G., 2005. State complexity of some operations on binary regular languages. Theor. Comput. Sci. 330 (2), 287–298.

[15] Maia, E., Moreira, N., Reis, R., 2013. Incomplete transition complexity of basic operations on finite languages. In: Konstantinidis, S. (Ed.), Proc. 18th CIAA. Vol. 7982 of LNCS. Springer, pp. 349–356.

[16] Maia, E., Moreira, N., Reis, R., 2013. Incomplete transition complexity of some basic operations. In: van Emde et al., P. (Ed.), Proc. 39th SOFSEM. Vol. 7741 of LNCS. Springer, pp. 319–331.

[17] Maurel, D., Guenthner, F., 2005. Automata and Dictionaries. College Publications.

[18] Owens, S., Reppy, J. H., Turon, A., 2009. Regular-expression derivatives re-examined. J. Funct. Program. 19 (2), 173–190.

[19] Salomaa, K., 2007. Descriptional complexity of nondeterministic finite automata. In: Harju, T., Karhumäki, J., Lepistö, A. (Eds.), Proc. 11th DLT. Vol. 4588 of LNCS. Springer, pp. 31–35.

[20] Salomaa, K., Yu, S., 1997. NFA to DFA transformation for finite languages over arbitrary alphabets. J. of Aut., Lang. and Comb. 2 (3), 177–186.

[21] Shallit, J., 2008. A Second Course in Formal Languages and Automata Theory. CUP.

[22] Yu, S., 1997. Regular languages. In: Rozenberg, G., Salomaa, A. (Eds.), Handbook of Formal Languages. Vol. 1. Springer, pp. 41–110.

[23] Yu, S., 2005. State complexity: Recent results and open problems. Fundam. Inform. 64 (1-4), 471–480.

[24] Yu, S., 2006. On the state complexity of combined operations. In: Ibarra, O. H., Yen, H.-C. (Eds.), Proc 11th CIAA. Vol. 4094 of LNCS. Springer, pp. 11–22.

[25] Yu, S., Gao, Y., 2011. State complexity research and approximation. In: Mauri, G., Leporati, A. (Eds.), Proc. 15th DLT. Vol. 6795 of LNCS. Springer, pp. 46–57.

[26] Yu, S., Zhuang, Q., Salomaa, K., 1994. The state complexities of some basic operations on regular languages. Theor. Comput. Sci. 125 (2), 315–328.