The Operational Incomplete Transition Complexity on Finite Languages

Eva Maia Nelma Moreira Rogério Reis
e-mail:{emaia,nam,rvr}@dcc.fc.up.pt
DCC-FC & CMUP, Universidade do Porto
Rua do Campo Alegre 1021, 4169-007 Porto, Portugal

Technical Report Series: DCC-2013-02
Version 1.3 April 2013
Old Versions:1.2 March 2013

1.1 February 2013

Departamento de Ciência de Computadores

Faculdade de Ciências da Universidade do Porto
Rua do Campo Alegre, 1021/1055,
4169-007 PORTO,
PORTUGAL

Tel: 220 402 900 Fax: 220 402 950
http://www.dcc.fc.up.pt/Pubs/

The Operational Incomplete Transition Complexity on Finite Languages

Eva Maia, Nelma Moreira, Rogério Reis

{emaia,nam,rvr}@dcc.fc.up.pt

DCC-FC & CMUP, Universidade do Porto

May 16, 2013

Abstract

The state complexity of basic operations on finite languages (considering complete DFAs) has been extensively studied in the literature. In this paper we study the incomplete (deterministic) state and transition complexity on finite languages of boolean operations, concatenation, star, and reversal. For all operations we give tight upper bounds for both descriptional measures. We correct the published state complexity of concatenation for complete DFAs and provide a tight upper bound for the case when the *right* automaton is larger than the *left* one. For all binary operations the tightness is proved using family languages with a variable alphabet size. In general the operational complexities depend not only on the complexities of the operands but also on other refined measures.

1 Introduction

Descriptional complexity studies the measures of complexity of languages and operations. These studies are motivated by the need to have good estimates of the amount of resources required to manipulate the smallest representation for a given language. In general, having succinct objects will improve our control on software, which may become smaller and more efficient. Finite languages are an important subset of regular languages with many applications in compilers, computational linguistics, control and verification, etc. [9, 1, 8, 3]. In those areas it is also usual to consider deterministic finite automata (DFA) with partial transition functions. As an example we can mention the manipulation of compact natural language dictionaries using Unicode alphabets. This motivates the study of the transition complexity of DFAs (not necessarily complete), besides the usual state complexity. The operational transition complexity of basic operations on regular languages was studied by Gao *et al.* [4] and Maia *et al.* [7]. In this paper we continue that line of research by considering the class of finite languages. For finite languages, Salomaa and Yu [10] showed that the state complexity of the determinization of a nondeterministic automaton (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.* [2] studied the operational state complexity of concatenation, Kleene star, and reversal. Finally, Han and Salomaa [5] gave tight upper bounds for the state complexity of union and intersection on finite languages. In this paper we give tight upper bounds for the state and transition complexity of all the above operations, for non necessarily complete DFAs with

Operation	Regular	$\|\Sigma\|$	Finite	$\|\Sigma\|$
$L_1 \cup L_2$	$2n(m+1)$	2	$\mathbf{3(mn\text{-}n\text{-}m)\ +2}$	$f_1(m,n)$
$L_1 \cap L_2$	nm	1	$\mathbf{(m-2)(n-2)(2+\sum_{i=1}^{\min(m,n)-3}(m-2-i)(n-2-i))+2}$	$f_2(m,n)$
L^C	$m+2$	1	$\mathbf{m+1}$	1
L_1L_2	$2^{n-1}(6m+3)-5$, if $m,n \geq 2$	3	$\mathbf{2^n(m-n+3)-8}$, if $m+1 \geq n$	2
			See Theorem 8 **(4)**	$n-1$
$L^\star$	$3.2^{m-1}-2$, if $m \geq 2$	2	$\mathbf{9 \cdot 2^{m-3} - 2^{m/2} - 2}$, if m is odd	3
			$\mathbf{9 \cdot 2^{m-3} - 2^{(m-2)/2} - 2}$, if m is even	
L^R	$2(2^m-1)$	2	$\mathbf{2^{p+2}-7}$, if $m=2p$	2
			$\mathbf{3 \cdot 2^p - 8}$, 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 each the upper bound is reached.

an alphabet size greater than 1 (see Table 2). For the concatenation, we correct the upper bound for the state complexity of complete DFAs [2], 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. The transition complexity results are all new, although the proofs are based on the ones for the state complexity and use techniques developed by Maia *et al.* [7]. Table 1 presents a comparison of the transition complexity on regular and finite languages, where the new results are highlighted. Note that the values in the table are obtained using languages for which the upper bounds are reached.

2 Preliminaries

We recall some basic notions about finite automata and regular languages. For more details, we refer the reader to the standard literature [6, 12, 11]. Given two integers $m, n \in \mathbb{N}$ let $[m,n] = \{i \in \mathbb{N} \mid m \leq i \leq n\}$. A *deterministic finite automaton* (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, $q_0 \in Q$ is the initial state, $F \subseteq Q$ is the set of final states, and δ is the transition function $Q \times \Sigma \to Q$. Let $|\Sigma| = k$, $|Q| = n$, and without lost of generality, we consider $Q = [0, n-1]$ with $q_0 = 0$. The transition function can be naturally extended to sets in 2^Q and to words $w \in \Sigma^\star$. A DFA is *complete* if the transition function is total. In this paper we consider DFAs to be not necessarily complete, *i.e.* with partial transition functions. The *language* accepted by A is $\mathcal{L}(A) = \{w \in \Sigma^\star \mid \delta(0, w) \in F\}$. Two DFAs are *equivalent* if they accept the same language. For each regular language there exists a unique minimal complete DFA with a least 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 $\equiv_L \subseteq \Sigma^\star \times \Sigma^\star$ is defined by $x \equiv_L y$ if and only if $D_x L = D_y L$. The *Myhill-Nerode Theorem* states that a language L is regular if and only if $\equiv_L$ has a finite number of equivalence classes, *i.e.*, L has a finite number of left quotients. This number is equal to the number of states of the minimal complete DFA, which is also the *state complexity* of L, denoted by $sc(L)$. If the minimal DFA is not complete its number of states is the number of left quotients minus one (the *dead state*, that we denote by Ω, is removed). The *incomplete state complexity* of a regular language L ($isc(L)$) is the number of states of the minimal DFA, not necessarily complete, that accepts L. Note that $isc(L)$ is either equal to $sc(L) - 1$ or to $sc(L)$. The *incomplete transition complexity*, $itc(L)$, of a regular language L is the minimal number of transitions over all DFAs that accepts L. We omit the term *incomplete* whenever the model is explicitly given. A τ-*transition* is a transition labeled by $\tau \in \Sigma$. The τ-*transition complexity* of L, $itc_\tau(L)$ is the minimal number of τ-transitions of any DFA recognizing L. It is known that $itc(L) = \sum_{\tau\in\Sigma} itc_\tau(L)$ [4, 7]. 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 are upper bounds 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 and called *witnesses*), such that the resulting language achieves that upper bound. For determining the transition complexity of an operation, we also consider the following measures and refined numbers of transitions. Let $A = ([0, n-1], \Sigma, \delta, 0, F)$ be a DFA, $\tau \in \Sigma$, and $i \in [0, n-1]$. We define $f(A) = |F|$, $f(A, i) = |F \cap [0, i-1]|$, $t_\tau(A, i)$ as 1 if exist a τ-transition leaving i and 0 otherwise, and $\bar{t}_\tau(a, i)$ as its complement. Let $s_\tau(A) = t_\tau(A, 0)$, $e_\tau(A) = \sum_{i\in F} t_\tau(A, i)$, $t_\tau(A) = \sum_{i\in Q} t_\tau(A, i)$, $t_\tau(A, [k, l]) = \sum_{i\in[k,l]} t_\tau(A, i)$, and the respective complements $\bar{s}_\tau(A) = \bar{t}_\tau(A, 0)$, $\bar{e}_\tau(A) = \sum_{i\in F} \bar{t}_\tau(A, i)$, etc. We denote by $in_\tau(A, i)$ the number of transitions reaching i, $a_\tau(A) = \sum_{i\in F} in_\tau(A, i)$ and $c_\tau(A, i) = 0$ if $in_\tau(A, i) > 0$ and 1 otherwise. Whenever there is no ambiguity we omit A from the above definitions. All the above measures, can be defined for a regular language L, considering the measure values for its minimal DFA. For instance, we have, $f(L)$, $f(L, i)$, $a_\tau(L)$, $e_\tau(L)$, etc. We define $s(L) = \sum_{\tau\in\Sigma} s_\tau(L)$ and $a(L) = \sum_{\tau\in\Sigma} a_\tau(L)$. Let A be a minimal DFA accepting a finite language, where the states are assumed to be topologically ordered. Then, $s(\mathcal{L}(A)) = 0$ and there is exactly one final state, denoted π and called *pre-dead*, such that $\sum_{\tau\in\Sigma} t_\tau(\pi) = 0$. The *level* of a state i is the size of the shortest path from the initial state to i, and never exceeds $n-1$. The level of A is the level of π. A DFA is *linear* if its level is $n-1$.

3 Union

Given two incomplete DFAs $A = ([0, m-1], \Sigma, \delta_A, 0, F_A)$ and $B = ([0, n-1], \Sigma, \delta_B, 0, F_B)$, let $C = (([0, m-1] \cup \{\Omega_A\}) \times ([0, m-1] \cup \{\Omega_B\})), \Sigma, \delta_C, (0, 0), (F_A \times ([0, m-1] \cup \{\Omega_B\})) \cup (([0, n-1] \cup \{\Omega_A\}) \times F_B))$ be a new DFA where for $\tau \in \Sigma$, $i \in [0, m-1] \cup \{\Omega_A\}$, and $j \in [0, n-1] \cup \{\Omega_B\}$,

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

It is not difficult to see that DFA C accepts the language $\mathcal{L}(A) \cup \mathcal{L}(B)$.

The two following theorems present upper bounds for the number of states and transitions for any DFA accepting $\mathcal{L}(A) \cup \mathcal{L}(B)$.

Theorem 1. *For any two finite languages L_1 and L_2 with $isc(L_1) = m$ and $isc(L_2) = n$, one has $isc(L_1 \cup L_2) \leq mn - 2$ and*

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

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

Proof. Let us consider the number of states. Considering the upper bound given by Han and Salomaa [5], $mn - m - n$ (see Table 3), we only need to remove the dead state of the resulting DFA. In the product automaton, the set of states is included in $([0, m-1] \cup \{\Omega_A\}) \times ([0, n-1] \cup \{\Omega_B\})$, where Ω_A and Ω_B are the dead states of the DFA A and DFA B, respectively. The states of the form $(0, i)$, where $i \in [1, n-1] \cup \{\Omega_B\}$, and of the form $(j, 0)$, where $j \in [1, m-1] \cup \{\Omega_A\}$, 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

$$\begin{aligned} & (m+1)(n+1) - (m+n) - 2 - 1 \\ = \;& mn - 2 \end{aligned}$$

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

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

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

3.1 Worst-case Witnesses

In the following we show that the upper bounds described above are tight. Han and Salomaa proved [5, 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 do not consider the dead state, our representation is slightly different. Let $m, n \geq 1$ and $\Sigma = \{b, c\} \cup \{a_{ij} \mid i \in [1, m-1], j \in [1, n-1],\ (i,j) \neq (m-1, n-1)\}$. Let $A = ([0, m-1], \Sigma, \delta_A, 0, \{m-1\})$ where $\delta_A(i, b) = i+1$ for $i \in [0, m-2]$ and $\delta_A(0, a_{ij}) = i$ for $j \in [1, n-1],\ (i,j) \neq (m-1, n-1)$. Let $B = ([0, n-1], \Sigma, \delta_B, 0, \{n-1\})$, where $\delta_B(i, c) = i+1$ for $i \in [0, n-1]$ and $\delta_B(0, a_{i,j}) = j$ for $j \in [1, n-1], i \in [1, m-1],\ (i,j) \neq (m-1, n-1)$.

See Figure 1 for the case $m = 5$ and $n = 4$.

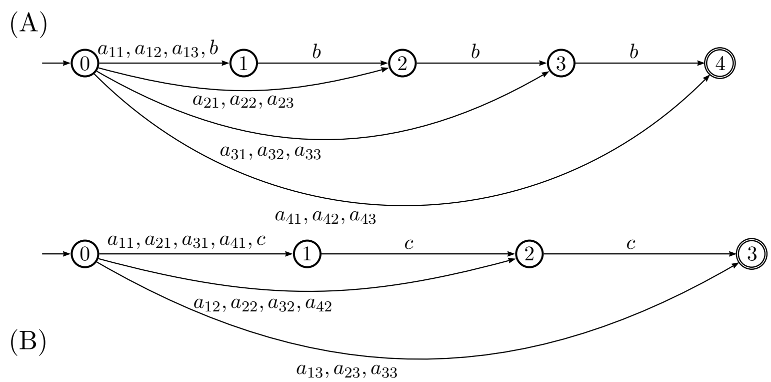

Figure 1: DFA A with $m = 5$ and DFA B with $n = 4$.

Theorem 2. *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 to the proof of [5, Lemma 2], considering the family languages above. Let us consider the number of transitions. The DFA A (Figure 1) has $m-1$ b-transitions and one a_{ij}-transition, for each a_{ij}. It has $mn - n - m$ different a_{ij}. The DFA B has $n-1$ c-transitions and the same number of a_{ij}-transitions as DFA A. Thus, the DFA resulting for the union operation has:

- $mn - 2n + 1$ b-transitions;
- $mn - 2n + 1$ c-transitions;
- 1 a_{ij}-transitions for each a_{ij}; 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 Intersection

Given two incomplete DFAs $A = ([0, m-1], \Sigma, \delta_A, 0, F_A)$ and $B = ([0, n-1], \Sigma, \delta_B, 0, F_B)$, a DFA accepting $\mathcal{L}(A) \cap \mathcal{L}(B)$ can be also obtained by the product construction, which leads to the following upper bounds for the number of states and transitions.

Theorem 3. *For any two finite languages L_1 and L_2 with $isc(L_1) = m$ and $isc(L_2) = n$, one has $isc(L_1 \cap L_2) \leq mn - 2m - 2n + 6$ and*

$$\begin{aligned} itc(L_1 \cap L_2) \leq \sum_{\tau \in \Sigma} & (s_\tau(L_1)s_\tau(L_2) + (itc_\tau(L_1) - s_\tau(L_1) - \\ & a_\tau(L_1))(itc_\tau(L_2) - s_\tau(L_2) - a_\tau(L_2)) + a_\tau(L_1)a_\tau(L_2)) . \end{aligned}$$

.

Proof. Consider the DFA accepting $\mathcal{L}(A) \cap \mathcal{L}(B)$ obtained by the product construction. For the same reasons as in Theorem 1, we can eliminate the states of the form $(0, j)$, where $j \in [1, n-1] \cup \{\Omega_B\}$, and of the form $(i, 0)$, where $i \in [1, m-1] \cup \{\Omega_A\}$; 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 $j \in [1, n-1] \cup \{\Omega_B\}$, and of the form (i, Ω_B), where $i \in [1, m-1] \cup \{\Omega_A\}$ 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

$$\begin{aligned} & (m+1)(n+1) - 3((m+1)(n+1)) + 12 - 1 \\ = \; & mn - 2m - 2n + 6 \\ = \; & mn - 2(m+n) + 6 \end{aligned}$$

Using the same technique as in Theorem 1 and considering that in the intersection we only have pairs of transitions where both elements are different from -1, the number of τ-transitions is as follows, which proves the theorem,

$$s_\tau(A)s_\tau(B) + (t_\tau(A, [1, m-1]) \setminus in_\tau(A, F_A))(t_\tau(B, [1, n-1]) \setminus in_\tau(B, F_B)) + a_\tau(A)a_\tau(B).$$

□

4.1 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 $\Sigma = \{a_{ij} \mid i \in [1, m-2],\ j \in [1, n-2]\} \cup \{a_{m-1,n-1}\}$. Let $A = ([0, m-1], \Sigma, \delta_A, 0, \{m-1\})$ where $\delta_A(x, a_{ij}) = x + i$ for $x \in [0, m-1],\ i \in [1, m-2]$, and $j \in [1, n-2]$, and let $B = ([0, n-1], \Sigma, \delta_B, 0, \{n-1\})$ where $\delta_B(x, a_{ij}) = x + j$ for $x \in [0, n-1],\ i \in [1, m-2]$, and $j \in [1, n-2]$. It is easy to see that A and B are minimal. The new families are presented in Figure 2 for $m = 5$ and $n = 4$.

Theorem 4. *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 [5, Lemma 6]. For that we will

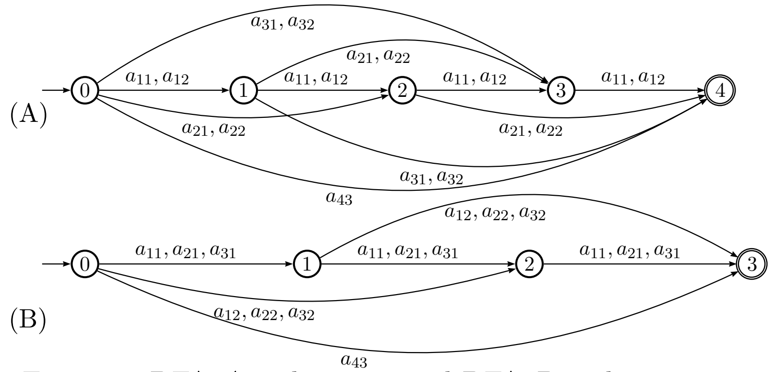

Figure 2: DFA A with $m = 5$ and DFA B with $n = 4$.

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:

$$\begin{aligned} R_1 &= \{\varepsilon\} \\ R_2 &= \{a_{m-1,n-1}\} \\ R_3 &= \{a_{ij} \mid i \in [1, m-2] \text{ and } j \in [1, n-2]\} \end{aligned}$$

It is easy to see that all words of each set are not equivalent to each other. The same is true with the words in R_3. As $|R_1| = |R_2| = 1$ and $|R_3| = (m-2)(n-2)$, we have that $|R| = mn - 2m - 2n + 6$ 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 3, 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 $2k + (m-2)(n-2)\sum_{i=1}^{\min(m,n)-3}(m-2-i)(n-2-i)$. As before the resulting DFA is minimal and the previous value corresponds exactly to the upper bound.

□

5 Complement

The state and transition complexity for this operation on finite languages are similar to the ones on regular languages [4, 7]. This happens because the DFA must be completed. Let $A = ([0, m-1], \Sigma, \delta_A, 0, F_A)$ be a DFA accepting the language L. The complement of L, L^c, is recognized by the DFA $C = ([0, m-1] \cup \{\Omega_A\}, \Sigma, \delta_C, 0, ([0, m-1] \setminus F) \cup \{\Omega_A\})$, where for $\tau \in \Sigma$ and $i \in [0, m-1]$, $\delta_C(i, \tau) = \delta_A(i, \tau)$ if $\delta_A(i, \tau) \uparrow$; $\delta_B(i, \tau) = \Omega_A$ otherwise. Therefore one has,

Theorem 5. *For any finite language L with $isc(L) = m$ one has $isc(L^C) \leq m + 1$ and $itc(L^C) \leq |\Sigma|(m + 1)$.*

Proof. It is easy to see that we only need to add the dead state. The maximal number of τ-transitions is $m+1$ because it is the number of states. Thus, the maximal number of transitions is $|\Sigma|(m+1)$. $\square$

Gao *et al.* [4] gave the value $|\Sigma|(itc(L)+2)$ 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 operands.

5.1 Worst-case Witnesses

The witness family for this operation is exactly the same presented in the referenced paper, i.e. $\{b^m\}$, for $m \geq 1$.

It is easy to see that the bounds are tight for this family (see Figure 3).

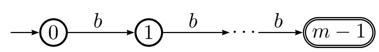

Figure 3: DFA A (accepting $\{b^m\}$) with m transitions and $m+1$ states.

6 Concatenation

Câmpeanu *et al.* [2] studied the state complexity of the concatenation of a m-state complete DFA A with a n-state complete DFA B over an alphabet of size k and proposed the upper bound

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

which was proved to be tight for $m > n-1$. It is easy to see that the second term of (1) 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[1] in Figure 5. This implies that (1) is not an upper bound if $m < n$. With these changes, we have

Theorem 6. *For any two finite languages L_1 and L_2 with $sc(L_1) = m$ and $sc(L_2) = n$ over an alphabet of size $k \geq 2$, one has*

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

Proof. The proof follows the one in [2] with the changes described above. $\square$

Given two incomplete DFAs $A = ([0, m-1], \Sigma, \delta_A, 0, F_A)$ and $B = ([0, n-1], \Sigma, \delta_B, 0, F_B)$, that represent finite languages, the algorithm by Maia *et al.* for the concatenation of regular languages can be applied to obtain a DFA $C = (R, \Sigma, \delta_C, r_0, F_C)$ accepting $\mathcal{L}(A)\mathcal{L}(B)$. The set of states of C is contained in the set $([0, m-1] \cup \{\Omega_A\}) \times 2^{[0,n-1]}$, the initial state r_0 is

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

$(0, \emptyset)$ if $0 \notin F_A$, and is $(0, \{0\})$ otherwise; $F_C = \{(i, P) \in R \mid P \cap F_B \neq \emptyset\}$, and for $\tau \in \Sigma$, $i \in [0, m-1]$, and $P \subseteq [0, n-1]$, $\delta_C((i, P), \tau) = (i', P')$ with $i' = \delta_A(i, \tau)$, if $\delta_A(i, \tau) \downarrow$ or $i' = \Omega_A$ otherwise, and $P' = \delta_B(P, \tau) \cup \{0\}$ if $i' \in F_A$ and $P' = \delta_B(P, \tau)$ otherwise. The next result is similar to the theorem above, omitting the dead state.

Theorem 7. *For any two finite languages L_1 and L_2 with $isc(L_1) = m$ and $isc(L_2) = n$ over an alphabet of size $k \geq 2$, one has*

$$isc(L_1L_2) \leq \sum_{i=0}^{m-1} \min\left\{k^i, \sum_{j=0}^{f(L_1,i)} \binom{n-1}{j}\right\} + \sum_{j=0}^{f(L_1)} \binom{n-1}{j} - 1. \tag{3}$$

Proof. Each state of the DFA C accepting $\mathcal{L}(A)\mathcal{L}(B)$ has the form (x, P) where $x \in [0, m-1] \cup \{\Omega_A\}$ and $P \subseteq [0, n-1]$. The first term of (3) corresponds to the maximal number of states of the form (i, P) with $i \in [0, m-1]$. Such a state (i, P) is at a level $\leq i$, 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 $\sum_{j=0}^{f(A,i)} \binom{n-1}{j}$. The number of states of the form (i, P) is the minimal of these two values. The second term of (3) corresponds to the maximal number of states with $x = \Omega_A$. In this case, the size of P is at most $f(A)$. Lastly, we remove the dead state. $\square$

For the transition complexity we have

Theorem 8. *For any two finite languages L_1 and L_2 with $isc(L_1) = m$ and $isc(L_2) = n$ over an alphabet of size k, and making $\Lambda_j = \binom{n-1}{j} - \binom{\overline{t}_\tau(L_2) - \overline{s}_\tau(L_2)}{j}$, $\Delta_j = \binom{n-1}{j} - \left(\binom{\overline{t}_\tau(L_2) - \overline{s}_\tau(L_2)}{j} * \overline{s}_\tau(L_2)\right)$ one has*

$$itc(L_1L_2) \leq k \sum_{i=0}^{m-2} \min\left\{k^i, \sum_{j=0}^{f(L_1,i)} \binom{n-1}{j}\right\} +$$
$$+ \sum_{\tau \in \Sigma} \left(\min\left\{k^{m-1} - \overline{s}_\tau(L_2), \sum_{j=0}^{f(L_1)-1} \Delta_j\right\} + \sum_{j=0}^{f(L_1)} \Lambda_j \right). \tag{4}$$

Proof. The τ-transitions of the DFA C accepting $\mathcal{L}(A)\mathcal{L}(B)$ have three forms: (i, β) where i represents the transition leaving the state $i \in [0, m-1]$; $(-1, \beta)$ where -1 represents the absence of the transition from state π_A 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 τ-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 $i \in [0, m-1]$ and $P \subseteq [0, n-1]$. The number of τ-transitions of the form $(-1, \beta)$ is $\min\{k^{m-1} - \overline{s}_\tau(L_2), \sum_{j=0}^{f(L_1)-1} \Delta_j\}$. We have at most k^{m-1} states in this level. However, if $s_\tau(B, 0) = 0$ 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 τ, if it exists. If this transition does not exist, *i.e.* $\overline{s}_\tau(B, 0) = 1$, we need to remove the sets with only non-defined transitions, because they originate transitions of the form $(-1, \emptyset)$. The number of τ-transitions of the form $(-2, \beta)$ is $\sum_{j=0}^{f(L_1)} \Lambda_j$ and this case is similar to the previous one. $\square$

6.1 Worst-case Witnesses

The next theorem characterizes the worst-case of the concatenation for finite languages.

Theorem 9. *The worst case for state and transition complexity of the concatenation of two incomplete DFAs is reached when the operands are linear.*

Proof. If the DFA is linear it has states in all levels. Thus, if the DFA is not linear it has for example a direct transition from the level i to the level $i+2$, so this DFA has less one state and less one transition than the linear one. $\square$

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

6.1.1 Case 1: $m+1 \geq n$

The witness languages are the ones presented by Câmpeanu *et al.* (see Figure 6).

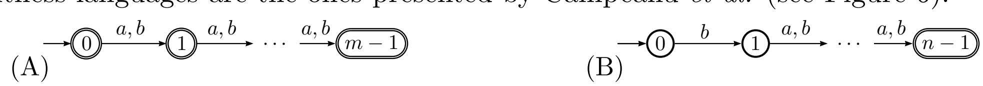

Figure 4: DFA A with m states and DFA B with n states.

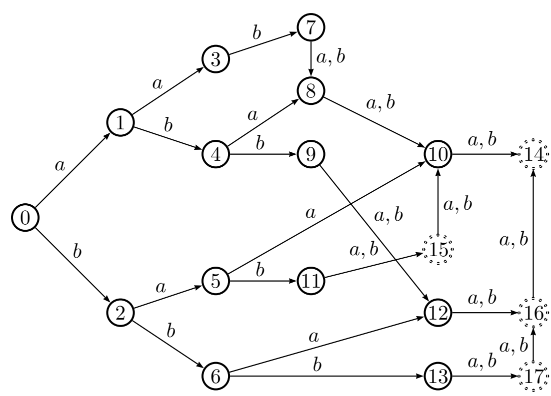

Figure 5: DFA resulting of the concatenation of DFA A with $m=3$ and DFA B with $n=5$, of Fig. 6. The states with dashed lines have level > 3 and are not accounted for by formula (1).

Theorem 10. *For any two integers $m \geq 2$ and $n \geq 2$ such that $m+1 \geq n$, 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. Fot the states the proof is similar to the proof of [2, Theorem 4]. Let $L = \mathcal{L}(C) = \mathcal{L}(A)\mathcal{L}(B)$. Consider all words $w_1, w_2 \in \Sigma^\star$, with $|w_1|, |w_2| \leq m-1$. If $|w_1| < |w_2|$ then $w_1 \not\equiv_L w_2$ since $b^{n+m-2-|w_1|} \in D_{w_1}L$ but $b^{n+m-2-|w_1|} \notin D_{w_2}L$. Let $|w_1| = |w_2|$ but $w_1 \neq w_2$. So w_1 and w_2 are different at the ith position from the right, for $i \leq n-1$. Assume that the ith position of w_1 is an a and the ith position of w_2 is a b. Then $w_1 \not\equiv_L w_2$ since $a^{n-1-i} \notin D_{w_1}L$ but $a^{n-1-i} \in D_{w_2}L$. For each $l \in [0, n-1]$, the words of size l belong to 2^l distinct equivalence classes of $\not\equiv_L$. For each $l \in [n-2, m-1]$, the words of size l belong to at least 2^{n-1} distinct equivalence classes of $\not\equiv_L$. Thus the number of equivalence classes of $\equiv_L$ are at least:

$$
\begin{aligned}
&1 + 2 + \ldots + 2^i + \ldots + 2^{n-1} + \underbrace{2^{n-2} + \ldots + 2^{n-2}}_{m-1-(n-1)+1 \text{ terms}} \\
&= (m - n + 3)2^{n-1} - 1
\end{aligned}
$$

We need to remove the class that corresponds to the dead state, thus we have $(m - n + 3)2^{n-1} - 2$.

The DFA A has $m-1$ τ-transitions for each $\tau \in \{a, b\}$. 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 8 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 (d, β). Thus,

$$
\begin{aligned}
& 2(2^{n-1}(m-n+1)-1) + 2^{n-1} - 2 + 2^{n-1} - 2 + 2^{n-1} + 2^{n-1} - 2 \\
= \; & 2^n(m-n+1) - 2 + 42^{n-1} - 6 \\
= \; & 2^n(m-n+3) - 8
\end{aligned}
$$

Therefore the theorem holds. □

6.1.2 Case 2: $m + 1 < n$

Let $\Sigma = \{b\} \cup \{a_i \mid i \in [1, n-2]\}$. Let $A = ([0, m-1], \Sigma, \delta_A, 0, [0, m-1])$ where $\delta_A(i, \tau) = i+1$, for any $\tau \in \Sigma$. Let $B = ([0, n-1], \Sigma, \delta_B, 0, \{n-1\})$ where $\delta_B(i, b) = i+1$, for $i \in [0, n-2]$, $\delta_B(i, a_j) = i + j$, for $i, j \in [1, n-2]$, $i + j \in [2, n-1]$, and $\delta_B(0, a_j) = j$, for $j \in [2, n-2]$. Note that A and B are minimal DFAs.

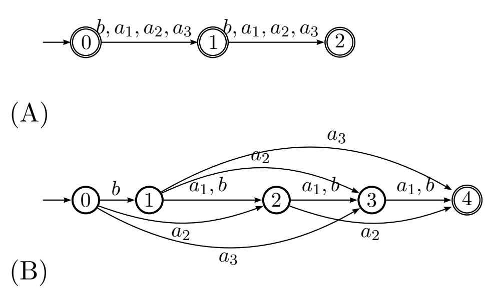

Figure 6: DFA A with $m = 3$ states and DFA B with $n = 5$ states.

Theorem 11. *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 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 $i \in [1, m-1]$ 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), $P \subseteq [0, n-1]$, $|P| \leq i+1$ and $0 \in P$, $i \in [1, m-1]$.
4. for each state (Ω_A, P), $\emptyset \neq P \subseteq [0, n-1]$, $|P| \leq m$ and $0 \notin P$.

Suppose that for a $1 \leq i \leq m-2$, 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,

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

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 an non empty set by construction of C. Let $l = \min(P')$ and $S_l = \{s - l \mid s \in P'\}$. Thus,

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

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 $\delta_C((i, P_2), a_1^{n-1-l}) \notin F_C$. The number of τ-transitions of DFA A is $m-1$, for $\tau \in \Sigma$. 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 8. □

Theorem 12. *The upper bounds for state and transition complexity of concatenation cannot be reached for any alphabet with a fixed size for $m \geq 0$, $n > m+1$.*

Proof. Let $S = \{(\Omega_A, P) \mid 1 \in P\} \subseteq R$. A state $(\Omega_A, P) \in S$ has to satisfy the following condition:

$$\exists i \in F_A \exists P' \subseteq 2^{[0,n-1]} \exists \tau \in \Sigma : \delta_C((i, P' \cup \{0\}), \tau) = (\Omega_A, P).$$

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 an absurd. The absurd resulted from supposing that k is fixed. □

7 Star

Given an incomplete DFA $A = ([0, m-1], \Sigma, \delta_A, 0, F_A)$ accepting a finite language, a DFA B accepting $\mathcal{L}(A)^\star$ can be constructed by an algorithm similar to the one for regular languages [7]. Let $B = (Q_B, \Sigma, \delta_B, \{0\}, F_B)$ where $Q_B \subseteq 2^{[0,m-1]}$, $F_B = \{P \in Q_B \mid P \cap F_A \neq \emptyset\} \cup \{0\}$, and for $\tau \in \Sigma$, $P \subseteq Q_B$, and $R = \delta_A(P, \tau)$, $\delta_B(P, \tau)$ is R if $R \cap F_A = \emptyset$, and $R \cup \{0\}$ otherwise.

If $f(A) = 1$ then the minimal DFA accepting $\mathcal{L}(A)^\star$ has also m states. Thus, we will consider DFAs with at least two final states. The following results give the number of states and transitions which are sufficient for any DFA B resulting from the previous algorithm.

Theorem 13. *For any finite language L with $isc(L) = m$ and $f(L) \geq 2$, one has $isc(L^\star) \leq 2^{m-f(L)-1} + 2^{m-2} - 1$ and*

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

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

Proof. The proof for the states is similar to the proof presented by Câmpeanu *et al.*. Note that in the star operation the states of the result DFA are sets of states of the DFA A. The minimal DFA accepting $L(A)^\star$ has at most the following states:

(i) the initial state 0_B which corresponds to the initial state of A; *i.e.*, 1 state

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

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

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

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

The proof for the transitions is similar to the one for the states. Enumerating the τ-transitions as done for the states, we have that:

(i) the presence or the absence of the transition leaving the initial state, $s_\tau(L)$;

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

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

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

Thus the theorem holds. □

7.1 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 dead state.

Let $A = ([0, m-1], \{a, b, c\}, \delta_A, 0, \{m-2, m-1\})$, $m \geq 4$, be a incomplete DFA accepting a finite language (see Figure 7) where:

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

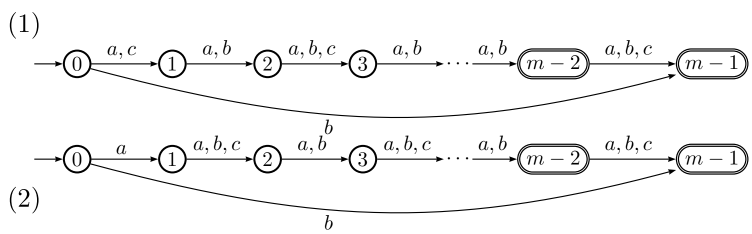

Figure 7: DFA A with m states, with m even (1) and odd (2).

Theorem 14. *For any integer $m \geq 4$ there exists 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 $9 \cdot 2^{m-3} - 2^{m/2} - 2$ transitions if m is odd, or $9 \cdot 2^{m-3} - 2^{(m-2)/2} - 2$ transitions otherwise.*

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 for transitions of Theorem 13 the DFA resulting for the star operation has:

- $3 \cdot 2^{m-3} - 1$ a-transitions.
- $3 \cdot 2^{m-3} - 1$ b-transitions.
- $3 \cdot 2^{m-3} - 2^{m/2}$ 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. ☐

8 Reversal

Given an incomplete DFA $A = ([0, m-1], \Sigma, \delta_A, 0, F_A)$, to obtain a DFA B that accepts $\mathcal{L}(A)^R$, we first reverse all transitions of A and then determinize the resulting NFA. Next we present upper bounds for the number of states and transitions of B.

Theorem 15. *For any finite languages L with $isc(L) = m$, $m \geq 3$, 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 $isc(L^R) \leq \sum_{i=0}^{l-1} k^i + 2^{m-l} - 1$ and if m is odd,*

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

or, if m is even,

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

Proof. The proof is similar to the proof of [2, Theorem 5]. We only need to remove the dead state. 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 set of transitions corresponding to the first $\sum_{i=0}^{l-1} k^i$ states and T_2 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 for the reversal are sets of states of DFA A we also consider each τ-transition as a set. If all τ-transitions were defined its number in T_2 would be 2^{m-l}. Note that the transitions of the $m - l$ states correspond to the transitions of the states between 0 and $l - 1$ in the initial DFA A, thus we remove the sets that has no τ-transitions. As the initial state of A has no transitions reaching it, we need to add one to the number of missing τ-transitions. Thus, $|T_2| = \sum_{\tau \in \Sigma} 2^{m-l} - 2^{(\sum_{i=0}^{l-1}(\bar{t}_\tau(i)))+1}$.

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

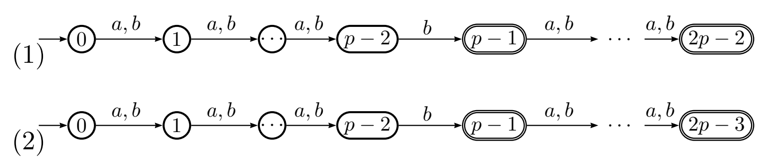

Figure 8: DFA A with $m = 2p - 1$ states (1) and with $m = 2p$ (2).

8.1 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 (see Figure 8).

Theorem 16. *For any integer $m \geq 4$ there exists 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.*. Considering the transitions as in the proof of Theorem **??** the DFA resulting for the reversal operation in case $m = 2p - 1$ has:

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

Thus, the resulting DFA in this case 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 the first set;
- $2^p - 2$ a-transitions in the second set;
- $2^{p-1} - 1$ a-transitions in the intermediate set;
- $2^p - 2$ b-transitions in the second set;
- 2^{p-1} b-transitions in the intermediate set;

Therefore the resulting DFA in this case has $2^{p+2} - 7$ transitions. $\square$

9 Final Remarks

In this paper we studied the incomplete state and transition complexity of basic regularity preserving operations on finite languages. Table 1 summarizes some of those results, using the witnesses parameters. Table 2 and Table 3 have the formulas for the upper bounds of state and transition complexity for these operations. 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. As future work we plan to study the average transition complexity of these operations.

References

[1] Beesley, K.R., Karttunen, L.: Finite State Morphology. CSLI Publications, Stanford University (2003)

[2] Câmpeanu, C., II, K.C., Salomaa, K., Yu, S.: State complexity of basic operations on finite languages. In: Boldt, O., Jürgensen, H. (eds.) WIA 1999. LNCS, vol. 2214, pp. 60–70. Springer (2001)

[3] Cassandras, C.G., Lafortune, S.: Introduction to discrete event systems. Springer (2006)

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

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

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

[7] Maia, E., Moreira, N., Reis, R.: Incomplete transition complexity of some basic operations. In: van Emde et al., P. (ed.) SOFSEM 2013. LNCS, vol. 7741, pp. 319–331. Springer (2013)

Operation	itc
$L_1 \cup L_2$	$\sum_{\tau\in\Sigma}(s_\tau(L_1)\boxplus s_\tau(L_2)-(itc_\tau(L_1)-s_\tau(L_1))(itc_\tau(L_2)-s_\tau(L_2)))\ +$ $n(itc(L_1)-s(L_1))+m(itc(L_2)-s(L_2))$
$L_1 \cap L_2$	$\sum_{\tau\in\Sigma}(s_\tau(L_1)s_\tau(L_2)+(itc_\tau(L_1)-s_\tau(L_1)$ $-\,a_\tau(L_1))(itc_\tau(L_2)-s_\tau(L_2)-a_\tau(L_2))+a_\tau(L_1)a_\tau(L_2))$
L^C	$\|\Sigma\|(m+1)$
L_1L_2	$k\sum_{i=0}^{m-2}\min\left\{k^i,\sum_{j=0}^{f(L_1,i)}\binom{n-1}{j}\right\}+$ $+\sum_{\tau\in\Sigma}\left(\min\left\{k^{m-1}-\overline{s}_\tau(L_2),\sum_{j=0}^{f(L_1)-1}\Delta_j\right\}+\sum_{j=0}^{f(L_1)}\Lambda_j\right)$
$L^\star$	$2^{m-f(L)-1}\left(k+\sum_{\tau\in\Sigma}2^{e_\tau(L)}\right)-\sum_{\tau\in\Sigma}2^{\overline{t}_\tau(L)-\overline{s}_\tau(L)-\overline{e}_\tau(L)}$ $-\sum_{\tau\in X}2^{\overline{t}_\tau(L)-\overline{s}_\tau(L)-\overline{e}_\tau(L)}$
L^R	$\sum_{i=0}^{l}k^i-1+k2^{m-l}-\sum_{\tau\in\Sigma}2^{\sum_{i=0}^{l-1}\overline{t}_\tau(L,i)+1}$, m even
	$\sum_{i=0}^{l}k^i-1+k2^{m-l}-\sum_{\tau\in\Sigma}\left(2^{\sum_{i=0}^{l-2}\overline{t}_\tau(L,i)+1}-c_\tau(l)\right)$, m odd

Table 2: Transition complexity of basic regularity preserving operations on finite languages

Operation	isc
$L_1 \cup L_2$	$mn-2$
$L_1 \cap L_2$	$mn-2m-2n+6$
L^C	$m+1$
L_1L_2	$\sum\limits_{i=0}^{m-1}\min\left\{k^i,\sum\limits_{j=0}^{f(A,i)}\binom{n-1}{j}\right\}+\sum\limits_{j=0}^{f(A)}\binom{n-1}{j}-1.$
$L^\star$	$2^{m-f(A)-1}+2^{m-2}-1$
L^R	$\sum_{i=0}^{l-1}k^i+2^{m-l}-1$

Table 3: State complexity of basic regularity preserving operations on finite languages

[8] Maurel, D., Guenthner, F.: Automata and Dictionaries. College Publications (2005)

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

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

[11] Shallit, J.: A Second Course in Formal Languages and Automata Theory. CUP (2008)

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