P. J. Safarik University

Faculty of Science

CONCATENATION OF REGULAR LANGUAGES AND STATE COMPLEXITY

ŠVK THESIS

Field of Study:	Informatika
Institute:	Ústav informatiky
Tutor:	RNDr. Juraj Šebej
Counsellor:	RNDr. Galina Jirásková, CSc.

Košice 2015 Erik Dorčák

Namiesto tejto strany vložte zadanie z informačného systému podpísané vedúcim ústavu!

Abstrakt

Zaoberáme sa stavovou zložitosťou zreťazenia. V práci uvádzame dôkaz pre dolnu hranicu zreťazenia. Ďalej sme sa zaoberali automatmi s polovicou koncových stavov. Pre tieto automaty uvádzame tiež dôkaz o dosahovaní hranice na nich, pričom tento výsledok bol motivovaný alternujúcimi automatmi, kde sa práve takéto stroje používajú na dôkaz dolnej hranice zreťazenia na alternujúcich strojoch.

Abstract

We study the state complexity of concatenation of regular languages represented by finite automata. We provide proof of lower bound of concatenation. Next we study automata with half of states final. For this class we prove tight bound in binary case. This result was motivated by work on alternating automata, where it is used to prove lower bound of concatenation of alternating automata.

Contents

Chapter 1

Introduction

Regular languages and finite automata are one of the oldest and the simplest topics in computer science. They have been investigated since the 1950s. Despite their simplicity, some problems are still open.

Motivating by applications of regular languages in software engineering, programming languages, and other areas in computer science, as well as by their importance in theory, this class of languages is intensively studied in recent years; for the discussion, we refer the reader to [3, 13]. Various areas in this field are now deeply and intensively examined. One of such areas is descriptional complexity which studies the cost of description of languages represented by different formal systems such as deterministic and nondeterministic finite automata, alternating and boolean automata, two-way automata, regular expressions, or grammars.

Rabin and Scott in 1959 [10] described an algorithm for the conversion of nondeterministic finite automata into deterministic automata known as the "subset construction". The algorithm shows that every n-state nondeterministic automaton can be simulating by at most 2^n state deterministic automaton. In 1963, Lupanov [6] proved the optimality of this construction by describing a ternary and even a binary regular language accepted by an n-state nondeterministic automaton that requires exactly 2^n deterministic states.

Maslov in 1970 [7] considered the state complexity of union, product, and Kleene star. He gave binary worst-case examples for these three operations, however he did not present any proofs. Birget in his work [1] examined intersection and union of several languages. The systematic study of the state complexity of operations on regular languages began in the paper by Yu, Zhuang, and Salomaa [14]. This work was followed by many papers studying state complexity of operations, until nowadays.

In this paper, we continue the study of the state complexity of concatenation of regular languages. In 1994, Yu *et al.* [14] provided upper bound for concatenation. They showed that $m2^n - k2^{n-1}$ states are sufficient for the DFA, which accepts $L \cdot K$, where DFA for L has m states and k finall states, and K has n states. Later in 2005 by Jirásková [5] was shown that this bound is tight in binary case.

In this paper we provide theorem about lower bound[5], where we present new proof. As this result was later used in [2]. But this application required more assumptions then provided by theorem. So next we are providing theorem, proof and automata which satisfies those assumptions, and therefore their results still holds true.

Chapter 2

Preliminaries

This section provides some basic definitions, notions and constructions which are used through this work. For more details and unlisted definitions and preliminary results, we refer to [11, 12].

We denote finite alphabet by Σ, by Σ^* we denote the set of all strings over alphabet Σ, including empty string denoted by ε. Let w be string, then $|w|$ means length of string w. A language is any subset of Σ^*.

We denote the size of a set A by $|A|$, and its power-set by 2^A.

A *deterministic finite state automaton* is a quintuple $A = (Q, \Sigma, \delta, s, F)$, where Q is a finite set of states; Σ is a finite alphabet; $\delta\colon Q \times \Sigma \to Q$ is the transition function, $s \in Q$ is the initial state; $F \subseteq Q$ is the set of final states (or accepting states). A non-final state q is a *dead state* if $\delta(q, a) = q$ for each a in Σ. The language accepted or recognized by the DFA A is defined to be the set $L(A) = \{w \in \Sigma^* \mid \delta(s, w) \in F\}$.

A *nondeterministic finite automaton* is a quintuple $A = (Q, \Sigma, \delta, s, F)$, where Q, Σ, s, and F are the same as for a DFA, and $\delta\colon Q \times \Sigma \to 2^Q$ is the transition function. Through the paper we use the notation (p, a, q) to mean that there is a transition from p to q on input a, that is, $q \in \delta(p, a)$. The language accepted by the NFA A is defined to be the set $L(A) = \{w \in \Sigma^* \mid \delta(s, w) \cap F \neq \emptyset\}$.

Two automata are *equivalent* if they recognize the same language.

A DFA A is *minimal* if every equivalent DFA has at least as many states as A. It is known that every regular language has a unique minimal DFA (up to isomorphism), and that a DFA $A = (Q, \Sigma, \delta, s, F)$ is minimal if and only if all its states are reachable and distinguishable.

The *state complexity* of a regular language L, denoted by $sc(L)$, is the number of states in the minimal DFA accepting the language L.

Every NFA can be converted to an equivalent DFA by the subset construction [11, 12] as follows. Let $A = (Q, \Sigma, \delta, s, F)$ be an NFA. Construct the DFA $A' = (2^Q, \Sigma, \delta', \{s\}, F')$, where $F' = \{R \subseteq Q \mid R \cap F \neq \emptyset\}$, and $\delta'(R, a) = \bigcup_{r \in R} \delta(r, a)$ for each R in 2^Q and each a in Σ. The DFA A' is called the *subset automaton* of the NFA A. The subset automaton may not be minimal since some of its states may be unreachable or equivalent.

For two regular languages K and L the *concatenation* $K \cdot L$ is defined to be $K \cdot L = \{uv \mid u \in K, v \in L\}$. For two DFA A, B with m, n states and A with k final states, we can construct NFA for concatenation of $L(A)L(B)$ with two constructions.

First construction using transitions on empty string ε, so called ε-acceptor. Let us take automatons A, B now we define new ε transitions from every final state of A which goes to initial state of B. All final states of A are non-final in NFA for concatenation and final states of NFA are only final states of B. Initial state of NFA is initial state of A.

Second construction define new transitions and possibly new initial states. Let us take automatons A, B now we define new transitions. For every state q of A and transition going on some symbol to final state of A, we add new transition from q by same symbol, going to initial state of B. If initial state of A is also final then initial states of NFA for concatenation are initial state of A and initial state B, so we get NFA with non-deterministic choice of initial state. Same as in previous construction, all final states of A are non-final in NFA for concatenation and final states of NFA are only final states of B.

Chapter 3

Upper bound of concatenation

The aim of this section is to show the tight bound on the state complexity of the concatenation operation on binary regular languages. Tight bound of concatenation was studied by Yu *et al.* [14] and by Jirásek *et al.* [4]. They showed that $m2^n - k2^{n-1}$ states are sufficient for the DFA, which accepts $L \cdot K$, where DFA for L has m states and k finall states, and K has n states. We recall their theorem on tight bound for binary alphabet, where we provide new proof. Yu also showed that in unary case bound is mn, and is tight when m and n are relatively prime. Unary case when m, n are not relatively prime was studied by Pighizziny nad Shallit in [8, 9].

We start with a lemma which provides strong argument for upper bound. This lemma show how many states can not be reached.

Lemma 3.1 *Let A, B be minimal automatons, F_A be set of final states of automaton A, and 0 initial state of B. Consider DFA D which is corresponding subset automaton to NFA for $A \cdot B$. Then every state of D, which contain $q \in F_A$, also contain state 0.*

Proof.
We separately show case when q is initial and final state of automaton A. Then automaton D has initial state $\{q, 0\}$. Let $q_j \in F_A$, $\sigma \in \Sigma$ and q_i be state of A which goes to q_j on σ. Depending on construction, there is ϵ transition from q_j to state 0, or there is transition from q_i which goes on σ to state 0. So when we reach state q_j, we also reach state 0. Therefore when q_j is in state of subset construction then there must be also state 0.

□

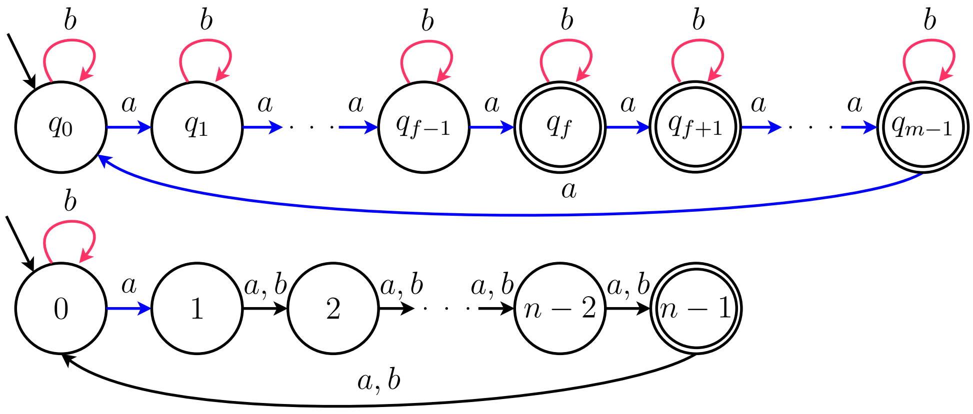

Figure 3.1: Examples of automata for tight bound. First automaton of concatenation is situated on the top of picture, second on the bottom.

Theorem 3.2 ([4]) *For any integers m, n, and k such that $m \geq 2, n \geq 2$ and $0 < k < m$, there exists a binary DFA A of m states and l accepting states, and a binary DFA B of n states such that any DFA accepting the language $L(A)L(B)$ needs at least $m2^n - k2^{n-1}$ states.*

Proof.

Let A be automaton shown in Fig. 3.1(top), and B be automaton shown in Fig. 3.1(bottom). Construct NFA C for concatenation of languages $L(A), L(B)$. We will prove that corresponding subset automaton to C has $m2^n - k2^{n-1}$ reachable, distinguishable states.

First, we show reachability of all $m2^n - k2^{n-1}$ states. The proof is by induction on the size of sets S, such that $S \subseteq \{0, ..., n-1\}$.

Assume three groups of states:

$I_x = \bigcup_{i=0}^{m-k-1} (q_i \cup S)$, where $|S| = x$,

$II_x = \bigcup_{i=m-k}^{m-1} (q_i \cup \{ 0 \} \cup S)$, where $|S| = x$,

$III_x = \bigcup_{i=0}^{m-k-1} (q_i \cup \{ 0 \} \cup S)$, where $|S| = x$. We also use $S \ominus x$, what mean subset of $\{0, \ldots, n-1\}$ which goes to S by a^x. As there are not two transition of $\{0, \ldots, n-1\}$ on a to same state this set is clearly determined.

1. Let $|S| = 0$.

   State q_0 is reachable because it is initial state of first automaton, thus it is initial state of automaton for concatenation. State q_i, such that $i \in \{0, ..., m-k-1\}$, is rechable from state q_0 by a^i. So all states of I_0 are reachable.

   Next we will show reachability of all states of II_0. State $q_{m-k} \cup \{0\}$ is reachable by a from q_{m-k-1}. Now assume other final states of first automaton with state 0.

That means that, we assume states q_i, where $i\,\{m-k, ..., m-1\}$, then $q_i \cup \{0\}$ is reachable from $q_{m-k} \cup \{0\}$ by $a^{i-(m-k)}b^n$.

We continue with rechability of states of III_x. State $q_0 \cup \{0\}$ is reachable from $q_{m-1} \cup \{0\}$ by ab^n. States $q_i \cup \{0\}$, where $i \in \{0, ..., m-k-1\}$, are reachable from $q_0 \cup \{0\}$ by $a^i b^n$. This completes reachability of III_0 and base case $x = 0$.

2. Let $|S| = x \geq 1$.
   Assume that we can reach all I_y, II_y, III_y where $y < x$. We will use induction hypothesis to prove reachability of I_x, II_x, III_x.

2.1. Reachability of I_x.
   Take $q_{m-1} \cup \{0\} \cup S \in II_{x-1}$, with $n-1 \notin S$. It goes to $q_0 \cup \{1\} \cup S_a$ by a. Next take $q_{i-1} \cup \{0\} \cup S \in III_{x-1}$, with $q_i \notin F_L$. Which goes to $q_i \cup \{1\} \cup S_a$ by a. By S_a we mean a set which is reached from a set S by a. We have all states of I_x such that $1 \in I_x$.

   Next we will prove reachability of states without 1, this means that, we want to reach $q_i \cup S$, where, $|S| = x$, $S \subseteq \{j, j+1, \ldots, n-1\}$ and $j \in S$. Assume $q_i \cup S'$ where $1 \in S'$ and $S' = S \ominus j$, so $q_i \cup S'$ is reachable. State $q_i \cup S$ is reachable from $q_i \cup S'$ by b^j. Now all states of I_x are reachable.

2.2. Reachability of II_x.
   We want to reach $q_{m-k} \cup \{0\} \cup S$. Let us take $q_{m-k-1} \cup S'$ from I_x such that $n-1 \notin S'$, $S' = S \ominus 1$. It goes by a to $q_{m-k} \cup \{0\} \cup S$. Now let us take $q_{i-1} \cup S'$, where $i \in \{m-k+1, \ldots, m-1\}$. Assume $q_{i-1} \cup S'$, with same S' as above, it goes by a to $q_i \cup \{0\} \cup S$. So all states of II_x are reachable.

1. Reachability of III_x.
   Let us take $q_{m-1} \cup \{0\} \cup \{n-1\} \cup S' \in II_x$, with $n-1 \in S'$. State $q_{m-1} \cup \{0\} \cup \{n-1\} \cup S'$ goes by a to $q_0 \cup \{0\} \cup S$, where $1 \in S$. Let i be smallest element of S. Until now we have arbitrary S with $i = 1$. To reach arbitrary S with $i \neq 1$, it is necessary to apply b^i to S', where $S' = S \ominus i$, $1 \in S'$.

   Next we use $q_0 \cup \{0\} \cup S'$ and $q_0 \cup S' \in I_x$ to reach $q_i \cup \{0\} \cup S$, where $i \in \{1, 2, \ldots, m-k-1\}$. State $q_i \cup \{0\} \cup S$, $i \in S$, can be reached from $q_0 \cup \{0\} \cup S'$, where $S' = S \ominus i$, by a^i. Or state $q_i \cup \{0\} \cup S$, $i \notin S$, can be reached from $q_0 \cup S'$, where $S' = S \ominus i$, by a^i. This completes reachability of all states of III_x.

Proof of reachability of all $m2^n - k2^{n-1}$ states is complete. Next we will prove distinguishability of all reachable states . Let S, T be subsets of $\{0, 1, \ldots, n-1\}$, and q_s, q_t be states of $\{q_0, q_1, \ldots q_{m-1}\}$. By appropriate choose of $q_s \cup S$ and $q_t \cup T$ we can subscribe every state of subset construction.

1. $S \neq T$.

   Without loss of generality, there exist i such that $i \in S$ and $i \notin T$. Therefore state $q_s \cup S$ goes by a^{n-1-i} to accepting state and $q_t \cup T$ by same string goes to rejecting state.

2. $S = T$ and $q_s \neq q_t$.

   Without loss of generality we assume that $s < t$. Define $q_{i\oplus 1}$ to be a state, which is reached from q_i by a. This case contain three subcases.

2.1. $q_{s\oplus 1},\ q_{t\oplus 1}\ \notin F_L$.

$q_s \cup S \xrightarrow{a^{m-k-t-1}} q_{s+m-k-t-1} \cup S'$, $q_{s+m-k-t-1} \notin F_L$, notice $q_{s+m-k-t} \notin F_L$.

$q_t \cup T \xrightarrow{a^{m-k-t-1}} q_{m-k-1} \cup T'$, $q_{m-k-1} \notin F_L$, notice $q_{m-k} \in F_L$.

Next we apply b^n:

If $S = T = \emptyset$, then:

$q_{s+m-k-t-1} \cup S' \xrightarrow{b^n} q_{s+m-k-t-1} \xrightarrow{a} q_{s+m-k-t}$

$q_{m-k-1} \cup T' \xrightarrow{b^n} q_{m-k-1} \xrightarrow{a} q_{m-k} \cup \{0\}$

On the other hand if S, T are non-empty:

$q_{s+m-k-t-1} \cup S' \xrightarrow{b^n} q_{s+m-k-t-1} \cup \{0\} \xrightarrow{a} q_{s+m-k-t} \cup \{1\}$

$q_{m-k-1} \cup T' \xrightarrow{b^n} q_{m-k-1} \cup \{0\} \xrightarrow{a} q_{m-k} \cup \{0, 1\}$

Both cases bring us to different subsets of $\{0, 1, \ldots, n-1\}$. That was considered in case 1.. Thus case is complete.

2.2. $q_{s\oplus 1} \notin F_L,\ q_{t\oplus 1} \in F_L$.

If $S = T = \emptyset$, then:

$q_s \cup S \xrightarrow{b^n} q_s \xrightarrow{a} q_{s\oplus 1}$

$q_t \cup T \xrightarrow{b^n} q_t \xrightarrow{a} q_{t\oplus 1} \cup \{0\}$

If S, T are non-empty:

$q_s \cup S \xrightarrow{b^n} q_s \cup \{0\} \xrightarrow{a} q_{s\oplus 1} \cup \{1\}$

$q_t \cup T \xrightarrow{b^n} q_t \cup \{0\} \xrightarrow{a} q_{t\oplus 1} \cup \{0, 1\}$

Again both cases bring us to different subsets of $\{0, 1, \ldots, n-1\}$. That was considered in case 1.. Thus case is complete.

2.3. $q_{s\oplus 1},\ q_{t\oplus 1} \in F_L$.

$q_s \cup S \xrightarrow{a^{m-1-t}} q_{m-1-t+s} \cup S'$

$q_t \cup T \xrightarrow{a^{m-1-t}} q_{m-1} \cup T'$

Let $q_{s'} = q_{m-1-t+s}$ a $q_{t'} = q_{m-1}$. Then we can denote $q_{t'\oplus 1} \notin F_L$, and $q_{s\oplus 1} \in F_L$. This bring us to case 2.2.

We showed that all states of subset automaton are pairwise distinguishable. Therefore we get minimal $m2^n - l2^{n-1}$ state automaton for concatenation. So our proof is complete.

□

Chapter 4

Automata with half final states

Previous chapter provided tight bound for concatenation with arbitrary amount of final states of first automaton. Paper [2] Chapter 2.1 An Application using mentioned result from [4] to prove lower bound on alternating finite automata (AFA). This result does not holds true when both automata have half of their states final, for example we can take $m = n = 4$. In this chapter we provide theorem, automata and proof which can be used in [2] as replacement. So their result that lower bound of concatenation of two AFA, with m and n states, is $2^m + n$ holds true.

Theorem 4.1 *For any even integers $m, n \geq 4$, there exists a binary DFA A of m states and $m/2$ accepting states, and a binary DFA B of n states and $n/2$ accepting states, such that any DFA accepting the language $L(A)L(B)$ needs $m2^n - m2^{n-2}$ states.*

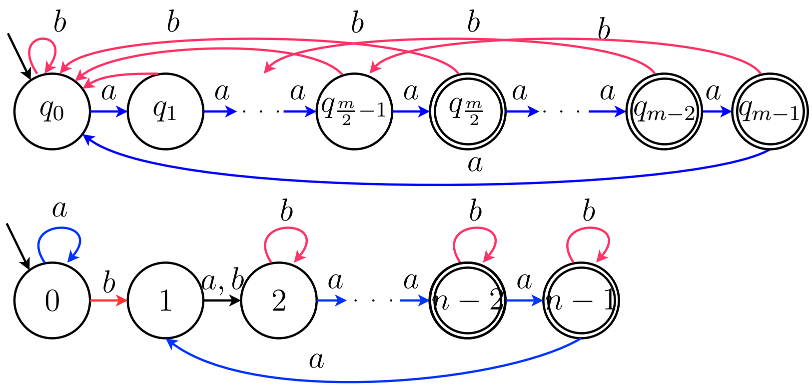

Figure 4.2: Examples of automata for tight bound with half of states final. First automaton of concatenation is situated on the top of picture, second on the bottom.

Proof.

Automata which satisfies assumptions of theorem are shown in Fig. 4.2. First we will prove reachability of $m2^n - m2^{n-2}$ states in subset automaton corresponding to concatenation of shown automata. The proof is by induction on the size of sets S, such that $S \subseteq \{1, ..., n-1\}$. Through the proof we use three notations for groups of states of subset automaton:

$I_k = \bigcup_{i=0}^{\frac{m}{2}-1} (q_i \cup S)$;

$II_k = \bigcup_{i=q_{\frac{m}{2}}}^{m-1} (q_i \cup \{0\} \cup S)$;

$III_k = \bigcup_{i=0}^{\frac{m}{2}-1} (q_i \cup \{0\} \cup S)$; where $|S| = k$. We also use $S \ominus x$, what mean subset of $\{1, \ldots, n-1\}$ which goes to S by a^x. As there are not two transition of $\{1, \ldots, n-1\}$ on a to same state this set is clearly determined.

Base case consist of showing reachability of I_0, II_0, III_0. Group of states I_0 are only non-final states of first automaton. State q_0 is reachable because it is initial state. Other states q_i, where $i \in \{1, 2, \ldots, \frac{m}{2} - 1\}$ are reachable by a^i from q_0. Group of states II_0 are final states of first automaton with state 0 from second automaton. Let q_i be final state of first automaton, that means $i \in \{\frac{m}{2}, \frac{m}{2} + 1, \ldots, m-1\}$. Let us take state $q_{\frac{m}{2}-1}$, which goes to $q_i, 0$ by $a^{i-\frac{m}{2}-1}$. Group of states III_0 are non-final states of first automaton with state 0 from second automaton. They are reached analogous as shown above. Assume states q_i, where q_i, where $i \in \{0, 1, \ldots, \frac{m}{2} - 1\}$. Then they can be reached from $q_{m-1}, 0$ by a^{i+1}. Base case is now complete.

Next we will prove reachability of groups I_k, II_k, III_k using induction hypothesis about reachability of all groups with less elements. Now we will prove reachability of I_k. Let us take set $q_F \cup \{0\} \cup S'$, which is from II_{k-1}, by q_F we mean some final state of first automaton. Consider four cases depending on states $1, 2$:

if $1, 2 \notin S'$, then $q_F \cup \{0\} \cup S' \xrightarrow{b} q_{F-\frac{m}{2}} \cup \{1\} \cup S', \quad S = S' \cup \{1\}$,

if $1 \in S', 2 \notin S'$, then $q_F \cup \{0\} \cup S' \xrightarrow{b} q_{F-\frac{m}{2}} \cup \{2\} \cup S', \quad S = S' \cup \{2\}$,

if $1 \notin S', 2 \in S'$, then $q_F \cup \{0\} \cup S' \xrightarrow{b} q_{F-\frac{m}{2}} \cup \{1\} \cup S', \quad S = S' \cup \{1\}$,

if $1, 2 \in S'$, then $q_F \cup \{0\} \cup S' \xrightarrow{b} q_{F-\frac{m}{2}} \cup S', \quad S = S'$.

Last case is mentioned only for sake of completeness, because size of S is not increased in fourth case. Also S is identical in second and third case. Until now we showed how to get S, such that S contains 1 or 2 or both of them.

Next we will show how to get S with smallest element bigger then 1. Let i be the smallest element of S, and $i \geq 2$. Then $q_0 \cup S$ can be reached by applying $(ab)^{i-1}$ on one of above mentioned cases, which contain 1; note that to get S, we need to start from $S \ominus (i-1)$. We get q_0 with arbitrary S. To get $q_i \cup S$ we need to apply string a^i to $q_0 \cup S \ominus i$.

We will use I_k to prove reachability of II_k. Let us take $q_{\frac{m}{2}-1} \cup S'$ from I_k, by $a^{i-(\frac{m}{2}-1)}$ it goes to $q_i \cup 0 \cup S$, where $S' = S \ominus (i - \frac{m}{2} - 1)$ for $i \in \left\{\frac{m}{2}, \ldots, m-1\right\}$.

Let us take $q_{m-1} \cup \{0\} \cup S'$ from II_k. It goes by a^{i+1} to $q_i \cup \{0\} \cup S$, where $i \in \left\{0, \ldots, \frac{m}{2} - 1\right\}$; note that as S' is necessary to take $S \ominus (i+1)$.

Finally, let: $I = \bigcup_{k=0}^{m-1} I_k$, $II = \bigcup_{i=0}^{m-1} II_k$, $III = \bigcup_{k=0}^{m-1} III_k$.
Then $|I| + |II| + |III| = \frac{m}{2}.2^{n-1} + \frac{m}{2}.2^{n-1} + \frac{m}{2}.2^{n-1} = \frac{3}{4}.m.2^n = m2^n - m2^{n-2}$, what completes proof about reachability.

Next we will prove distinguishability of all reachable states by finding unique accepted string for every state of NFA for concatenation. That means finding string such that it is accepted only from this state but is rejected from every other string. Therefore every reachable state of subset construction is distinguishable from other states.

We start with state 2. Let w_2 be the unique string for state the 2, it is following:

$$w_2 = (\prod_{i=0}^{n-4} a^{n-3-i}bba^{i+2})a^{n-k-2}$$

. Next we will analyse string w_2. Every state q_i goes by every $a^{n-3-i}bba^{i+2}$ to some state of first automaton or after bb it is in state 2 from which it goes to $2+i+2$. State 0 goes again to $2+i+2$. States i, where $i \in \{1, \ldots, n-1\}$ goes by $a^{n-3-i}bba^{i+2}$ itself if $i \neq i+3$, and to itself plus one otherwise. So after applying product part of w_2, on all states we get some state from first part together with $1, 2$, all states except 2 went to state 1, only state 2 went to itself. Now it is easy to show how to bring state 2 into final state and 1 into non-final state, what can be done by string a^{n-k-2}.

Next we will show unique words how to get into state 2 from which we can continue with $w = 2$. As states i, where $i \in \{1, 3, 4, \ldots, n-1\}$, i goes to state 2 by $a^{n-1+2-i}$. State 0 goes to 2 by ba.

For the states of first automaton we will show string for q_{m-1} and then unique string how to get to q_{m-1}. State q_{m-1} goes to 2 by $baba$. Every other state q_i where $i \in 0, 1, \ldots, m-2$, goes to q_{m-1} by a^{m-1-i}.

This completes proof distinguishability, therefore our proof is complete.

□

Bibliography

[1] Birget, J.-C.: Intersection and union of regular languages and state complexity. Inform. Process. Lett. 43, 185–190 (1992)

[2] Čevorová, K., Jirásková, G., Krajnáková, I., K.: On the Square of Regular Languages. CIAA 2014: 136-147 (2014)

[3] Hromkovič, J.: Descriptional complexity of finite automata: Concepts and open problems. J. Autom. Lang. Comb. 7, 519–531 (2002)

[4] Jirásek, J., Jirásková, G., Szabari, A.: State complexity of concatenation and complementation. Int. J. Found. Comput. Sci. 16(3): 511-529 (2005)

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

[6] Lupanov, U.I.: A comparison of two types of finite automata. Problemy Kibernetiki 9, 321-326 (1963) (in Russian)

[7] Maslov, A.N.: Estimates of the number of states of finite automata. Soviet Math. Dokl. 11, 1373–1375 (1970)

[8] Pighizzini, G.,: Unary language concatenation and its state complexity, in: S. Yu, A. Pun (Eds.), Implementation and Application of automata: 5th International Conference, CIAA 2000, LNCS 2088, Springer-Verlag, 2001, pp. 252–262 (2000)

[9] Pighizzini, G.,: Unary language operations, state complexity and Jacobsthal's function, Internat. J. Found. Comput. Sci. 13 145–159 (2002)

[10] Rabin, M., Scott, D.: Finite automata and their decision problems. IBM Res. Develop. 3, 114–129 (1959)

[11] Sipser, M.: Introduction to the theory of computation. PWS Publishing Company, Boston (1997)

[12] Yu, S.: Chapter 2: Regular languages. In: Rozenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages - Vol. I, pp. 41–110. Springer, Heidelberg (1997)

[13] Yu, S.: A renaissance of automata theory? Bull. Eur. Assoc. Theor. Comput. Sci. 72, 270–272 (2000)

[14] Yu, S., Zhuang, Q., Salomaa, K.: The state complexity of some basic operations on regular languages. Theoret. Comput. Sci. 125, 315–328 (1994)