

Some more algorithms for Conway’s universal automaton

Boris MELNIKOV

Russia, Togliatti State University
email: B.Melnikov@tltsu.ru

Vasily DOLGOV

Russia, Togliatti State University
email: terenga74@mail.ru

Abstract. In this paper authors propose algorithms for constructing so-called automaton $\mathcal{COM}(L)$ and prove that this automaton coincides, up to re-denoting states, with Conway’s universal automaton. We give some algorithms of constructing this automaton and consider some examples.

1 Introduction

In this paper authors propose algorithms for constructing so-called automaton $\mathcal{COM}(L)$. The definition of this automaton could be constructed by [8]. Below, we shall consider this definition. We also prove that this automaton coincides, up to re-denoting states, with Conway’s universal automaton \mathcal{U}_L ([4] etc.). We hope that in future it will be possible to show that “in average” (for specially defined “average” notion) this algorithms are more effective than algorithms which construct \mathcal{U}_L by definition, i.e., using (sub)factorisations etc.

Although we do not consider classic complexity problems here, it is obvious that for constructing \mathcal{U}_L (Section 5) complexity is proportional to the number of grids, which is considered in Section 7.

The contents of this article is as follows.

In Section 2 we introduce some notions: binary relation $\#$, pseudo-grid, grid, covering subset of the grids. This notions appeared in [5, 6].

Computing Classification System 1998: F.4.3

Mathematics Subject Classification 2010: 68Q45

Key words and phrases: nondeterministic finite automata, Conway’s universal automaton, constructing algorithms

DOI:10.2478/ausi-2014-0015

In Section 3 we give definition of automaton $\mathcal{COM}(L)$ and prove that this automaton coincides with \mathcal{U}_L .

In Section 4 we define “covering automata” by automaton $\mathcal{COM}(L)$.

In Section 5 we give two algorithms of constructing automaton \mathcal{U}_L , the second of them being “the mirror image” of the first. This algorithms rely on the algorithm, described in [6]. Also we give a simplified version of the first algorithm.

In Section 6 we give a detailed example of the work of algorithms, described in the previous section. An example of *equivalent* covering automaton is also given.

In Section 7 we consider examples which shows how fast a number of grids (i.e., the size of the universal automaton) can grow if we are given the sizes of two canonical automata (i.e., \tilde{L} and \tilde{L}^R).

In Section 8 we give some ideas for further research.

2 Preliminaries

We shall use designations of [5, 6]. Let us repeat the main of them.

The language of nondeterministic finite automaton

$$K = (Q, \Sigma, \delta, S, F)$$

will be defined by $\mathcal{L}(K)$. For a state q of this automaton, we shall denote the language of automaton $(Q, \Sigma, \delta, S, \{q\})$ by $\mathcal{L}_K^{in}(q)$. \tilde{L} is the canonical automaton defining regular language L , we shall consider canonical automata without the useless (“dead”) state. Let automata \tilde{L} and \tilde{L}^R for the given language L be as follows:

$$\tilde{L} = (Q_\pi, \Sigma, \delta_\pi, \{s_\pi\}, F_\pi) \quad \text{and} \quad \tilde{L}^R = (Q_\rho, \Sigma, \delta_\rho, \{s_\rho\}, F_\rho).$$

Binary relation $\# \subseteq Q_\pi \times Q_\rho$ is defined in the following way. For some states $A \in Q_\pi$ and $X \in Q_\rho$, condition $A\#X$ holds if and only if there exist some words $u \in \mathcal{L}_L^{in}(A)$ and $v \in \mathcal{L}_{L^R}^{out}(X)$, such that $uv^R \in \mathcal{L}(K)$. In [6], we considered a simple algorithm for constructing this relation.

If for some pair $P \subseteq Q_\pi$ and $R \subseteq Q_\rho$ we have

$$(\forall A \in P) (\forall X \in R) (A\#X),$$

then $\mathcal{B} = (P, R)$ is a pseudo-grid. For it, we shall write $\alpha(\mathcal{B}) = P$ and $\beta(\mathcal{B}) = R$.

If for some pseudo-grid $\mathcal{B} = (P, R)$, there exists

- neither $A \in Q_\pi \setminus P$ such that $((P \cup \{A\}), R)$ is also a pseudo-grid,
- nor $X \in Q_\rho \setminus R$ such that $(P, (R \cup \{X\}))$ is also a pseudo-grid,

then \mathcal{B} is a grid.

For the given regular language L , we shall consider its set of grids; in the next sections we shall denote it by $Q_{\mathcal{COM}}$. Some its subset $\mathcal{Q} \subseteq Q_{\mathcal{COM}}$ will be called by *covering* subset of the grids, if for any $A \in Q_\pi$ and $X \in Q_\rho$ such that $A \# X$, there exists a grid $\mathcal{B} \in \mathcal{Q}$, such that $A \in \alpha(\mathcal{B})$ and $X \in \beta(\mathcal{B})$.¹

As we said before, we shall also use the universal automaton for the given regular language L ; by [4, Def.2.4], we shall denote it by \mathcal{U}_L . However, its elements we shall mark by subscripts \mathcal{U}_L ; e.g., its transition function will be denoted by $\delta_{\mathcal{U}_L}$.

3 Definition of automaton $\mathcal{COM}(L)$ (an alternative definition of automaton \mathcal{U}_L)

Automaton $\mathcal{COM}(L)$ for the given regular language L has to be defined in the usual way, i.e., by a quintet. For now, we only have the set of states $Q_{\mathcal{COM}}$ (as we said before, it is the set of grids for the given *language*) and alphabet Σ ; let us define sets of initial and final states and the transition function.

Thus, considering automata \tilde{L} and \tilde{L}^R , we define automaton

$$\mathcal{COM}(L) = (Q_{\mathcal{COM}}, \Sigma, \delta_{\mathcal{COM}}, S_{\mathcal{COM}}, F_{\mathcal{COM}}),$$

where:

- $S_{\mathcal{COM}} = \{ \mathcal{B} \in Q_{\mathcal{COM}} \mid \alpha(\mathcal{B}) \ni s_\pi \}$;
- $F_{\mathcal{COM}} = \{ \mathcal{B} \in Q_{\mathcal{COM}} \mid \beta(\mathcal{B}) \ni s_\rho \}$;
- for some pair $\mathcal{B}_1, \mathcal{B}_2 \in Q_{\mathcal{COM}}$ (condition $\mathcal{B}_1 = \mathcal{B}_2$ is possible) and some $\mathbf{a} \in \Sigma$, we set

$$\delta_{\mathcal{COM}}(\mathcal{B}_1, \mathbf{a}) \ni \mathcal{B}_2 \quad (\text{i.e., } \mathcal{B}_1 \xrightarrow[\delta_{\mathcal{COM}}]{\mathbf{a}} \mathcal{B}_2)$$

if and only if both the following conditions hold:

$$\left(\forall A \in \alpha(\mathcal{B}_1) \right) \left((\delta_\pi(A, \mathbf{a}) \neq \emptyset) \ \& \ (\delta_\pi(A, \mathbf{a}) \subseteq \alpha(\mathcal{B}_2)) \right); \quad (1)$$

$$\left(\forall Y \in \beta(\mathcal{B}_2) \right) \left((\delta_\rho(Y, \mathbf{a}) \neq \emptyset) \ \& \ (\delta_\rho(Y, \mathbf{a}) \subseteq \beta(\mathcal{B}_1)) \right). \quad (2)$$

¹ We shall not consider *algorithms* for constructing such subsets.

Let us remark, that considering canonical automaton *having* possible “dead” state, we obtain *the same* values of φ^{in} and φ^{out} (because these values cannot contain these “dead” states). Then the definition of $\mathcal{COM}(L)$ for the given language L is independent of considering “dead” state.

Also let us remark, that we can write (1) in the following way:

$$\left(\forall A \in \alpha(\mathcal{B}_1) \right) \left((\delta_\pi(A, a) = \{B\}) \ \& \ (B \in \alpha(\mathcal{B}_2)) \right)$$

(similarly for (2)). But we *cannot* write it in the following way:

$$\left(\forall A \in \alpha(\mathcal{B}_1) \right) \left(\delta_\pi(A, a) \subseteq \alpha(\mathcal{B}_2) \right) \quad (3)$$

(because the value of $\delta_\pi(A, a)$ can be \emptyset). But considering canonical automaton *having* possible “dead” state (such an automaton is total) we can write it by (3). And considering canonical automaton having transition function of the type

$$\delta : Q \times \Sigma \rightarrow Q$$

(like, for example, [1]), we can write (3) in the following simple way:

$$\left(\forall A \in \alpha(\mathcal{B}_1) \right) \left(\delta_\pi(A, a) \in \alpha(\mathcal{B}_2) \right).$$

The following theorem formulates the correctness of both the definitions given before.

Theorem 1 $\mathcal{COM}(L) = \mathcal{U}_L$.²

Proof. Firstly, let us prove for each state $\mathcal{B} \in Q_{\mathcal{COM}}$, that the pair

$$\left(\mathcal{L}_{\mathcal{COM}}^{in}(\mathcal{B}), \mathcal{L}_{\mathcal{COM}}^{out}(\mathcal{B}) \right) \quad (4)$$

is a factorization of L . For it let us suppose, that (4) is only a subfactorization of L (not a factorization). Then we would have:

- either some word $u \in \Sigma^*$ (where $u \notin \mathcal{L}_{\mathcal{COM}}^{in}(\mathcal{B})$), such that

$$\left(\mathcal{L}_{\mathcal{COM}}^{in}(\mathcal{B}) \cup \{u\}, \mathcal{L}_{\mathcal{COM}}^{out}(\mathcal{B}) \right)$$

is also a subfactorization (or a factorization) of L ;

² Up to re-denoting states.

- or some word $v \in \Sigma^*$ (where $v \notin \mathcal{L}_{\omega\mathcal{M}}^{out}(\mathcal{B})$), such that

$$\left(\mathcal{L}_{\omega\mathcal{M}}^{in}(\mathcal{B}), \mathcal{L}_{\omega\mathcal{M}}^{out}(\mathcal{B}) \cup \{v\} \right)$$

is also a subfactorization (or a factorization) of L .

Without loss of generality, we shall consider the first case. ³

Besides, we would have a grid $\mathcal{B}' \in Q_{\omega\mathcal{M}}$, for which

$$\mathcal{L}_{\omega\mathcal{M}}^{in}(\mathcal{B}') \ni \mathcal{L}_{\omega\mathcal{M}}^{in}(\mathcal{B}) \cup \{u\}.$$

I.e., for the state A of automaton \tilde{L} , such that $\mathcal{L}_{\tilde{L}}^{in}(A) \ni u$, we would have that

$$A \# X \quad \text{for each } X \in \beta(\mathcal{B}).$$

Then $(\alpha(\mathcal{B}) \cup \{A\}) \times \beta(\mathcal{B})$ is a pseudo-grid (or a grid, see [6]), and for \mathcal{B} , we obtain a contradiction with the definition of the grid.

Vice versa, let $(\mathcal{X}, \mathcal{Y})$ be a state of automaton \mathcal{U}_L . Consider the sets $P \subseteq Q_\pi$ and $R \subseteq Q_\rho$ defined in the following way:

$$P = \left\{ A \in Q_\pi \mid (\exists u \in \mathcal{X}) (\mathcal{L}_{\tilde{L}}^{in}(A) \ni u) \right\},$$

$$\text{and } R = \left\{ X \in Q_\rho \mid (\exists v \in \mathcal{Y}^R) (\mathcal{L}_{\tilde{L}^R}^{in}(X) \ni v) \right\}.$$

Since $(\mathcal{X}, \mathcal{Y})$ is a factorization of L , then for each pair of states $A \in P$ and $X \in Q$, we have $A \# X$. Therefore $P \times Q$ is a pseudo-grid.

And if $P \times Q$ is *not* a grid, then we would add some words to \mathcal{X} or \mathcal{Y} satisfying the definition of (sub) factorization; then $(\mathcal{X}, \mathcal{Y})$ would be not a factorization.

Let us prove the coincidence of the sets of edges. By the definition of \mathcal{U}_L ,

$$(\mathcal{X}', \mathcal{Y}') \in \delta_{\mathcal{U}_L}((\mathcal{X}, \mathcal{Y}), a) \quad \text{holds if } \mathcal{X}' a \mathcal{Y}' \subseteq L.$$

The same condition holds also for $\delta_{\omega\mathcal{M}}$, i.e.,

$$\mathcal{B}' \in \delta_{\omega\mathcal{M}}(\mathcal{B}, a) \quad \text{holds if } \mathcal{L}_{\omega\mathcal{M}}^{in}(\mathcal{B}) a \mathcal{L}_{\omega\mathcal{M}}^{out}(\mathcal{B}') \subseteq L,$$

where \mathcal{B}' corresponds to $(\mathcal{X}', \mathcal{Y}')$ and \mathcal{B} corresponds to $(\mathcal{X}, \mathcal{Y})$. □

³ (Sub) factorization $(\mathcal{L}_{\omega\mathcal{M}}^{in}(\mathcal{B}) \cup \{u\}, \mathcal{L}_{\omega\mathcal{M}}^{out}(\mathcal{B}) \cup \{v\})$ is also possible.

4 Covering automata

Using automaton $\mathcal{COM}(L)$ and *the given* covering subset of the grids $\mathcal{Q} \subseteq \mathcal{Q}_{\mathcal{COM}}$, let us also define corresponding *covering automaton*. We define it in the following way:

$$\mathcal{COM}_{\mathcal{Q}}(L) = (\mathcal{Q}, \Sigma, \delta_{\mathcal{Q}}, S_{\mathcal{Q}}, F_{\mathcal{Q}}),$$

where:

- $S_{\mathcal{Q}} = \mathcal{Q} \cap S_{\mathcal{COM}}$;
- $F_{\mathcal{Q}} = \mathcal{Q} \cap F_{\mathcal{COM}}$;
- $\delta_{\mathcal{Q}} = \left\{ \mathcal{B}_1 \xrightarrow{a} \mathcal{B}_2 \mid a \in \Sigma, \mathcal{B}_1, \mathcal{B}_2 \in \mathcal{Q}, \mathcal{B}_1 \xrightarrow[\delta_{\mathcal{COM}}]{a} \mathcal{B}_2 \right\}$.

For now, we do not consider the equivalence of automata $\mathcal{COM}(L)$ and $\mathcal{COM}_{\mathcal{Q}}(L)$ (i.e., whether automaton $\mathcal{COM}_{\mathcal{Q}}(L)$ defines the given language L); some examples will be considered in the next sections.

5 Algorithms for constructing automaton \mathcal{U}_L

In [6], we considered a possible algorithm of constructing automaton \widetilde{L}^R ; using this algorithm, we obtained *at the same time* values of functions φ^{in} and φ^{out} and binary relation $\#$. In this section, we shall obtain automaton \mathcal{U}_L using *the same algorithm*.

Thus, let us suppose that we already have all these objects. Simply by definitions of grids and automaton $\mathcal{COM}(L)$, and also by Theorem 1, we obtain considering all the subsets of the set \mathcal{Q}_{π} the following

Algorithm 1 (Constructing automaton \mathcal{U}_L)

Input: automata \widetilde{L} , \widetilde{L}^R , binary relation $\#$.

Output: automaton \mathcal{U}_L .

Step 1. Consider array $U[\text{index}]$, where index can be each element of $\mathcal{P}(\mathcal{Q}_{\pi})$ (except \emptyset), and its values can be elements of $\mathcal{P}(\mathcal{Q}_{\rho})$. For each possible index , we set

$$U[\text{index}] := \bigcap_{A \in \text{index}} \left\{ X \in \mathcal{Q}_{\rho} \mid A \# X \right\}.$$

Step 2. Consider Boolean array $B[\text{index}]$, where index can be each element of $\mathcal{P}(\mathcal{Q}_{\pi})$ (except \emptyset). For each possible index , we set

$$B[\text{index}] := (U[\text{index}] \neq \emptyset).$$

Step 3. For each possible \mathbf{index} , such that condition $\mathbf{B}[\mathbf{index}]$ holds, if

$$(\exists \mathbf{ind} \in \mathcal{P}(Q_\pi)) ((\mathbf{index} \subset \mathbf{ind}) \ \& \ (\mathbf{U}[\mathbf{ind}] = \mathbf{U}[\mathbf{index}])),$$

then we set $\mathbf{B}[\mathbf{index}] := \mathbf{false}$.

Step 4. We select the following set of grids:

$$Q_{\mathcal{COM}} = \left\{ \mathbf{index} \times \mathbf{U}[\mathbf{index}] \mid \mathbf{index} \in \mathcal{P}(Q_\pi) \ \& \ \mathbf{B}[\mathbf{index}] \right\}.$$

Step 5. $\delta_{\mathcal{COM}}$, $S_{\mathcal{COM}}$ and $F_{\mathcal{COM}}$ are defined by definition of automaton $\mathcal{COM}(\mathbb{L})$ given before. \square

Let us formulate the “mirror image” of this algorithm, where we at first consider subsets of Q_ρ .

Algorithm 2 (Constructing automaton \mathcal{U}_L)

Input: automata \widetilde{L} , \widetilde{L}^R , binary relation $\#$.

Output: automaton \mathcal{U}_L .

Step 1. Consider array $\mathbf{U}[\mathbf{index}]$, where \mathbf{index} can be each element of $\mathcal{P}(Q_\rho)$ (except \emptyset), and its values can be elements of $\mathcal{P}(Q_\pi)$. For each possible \mathbf{index} , we set

$$\mathbf{U}[\mathbf{index}] := \bigcap_{X \in \mathbf{index}} \left\{ A \in Q_\pi \mid A \# X \right\}.$$

Step 2. Consider Boolean array $\mathbf{B}[\mathbf{index}]$, where \mathbf{index} can be each element of $\mathcal{P}(Q_\rho)$ (except \emptyset). For each possible \mathbf{index} , we set

$$\mathbf{B}[\mathbf{index}] := (\mathbf{U}[\mathbf{index}] \neq \emptyset).$$

Step 3. For each possible \mathbf{index} , such that condition $\mathbf{B}[\mathbf{index}]$ holds, if

$$(\exists \mathbf{ind} \in \mathcal{P}(Q_\rho)) ((\mathbf{index} \subset \mathbf{ind}) \ \& \ (\mathbf{U}[\mathbf{ind}] = \mathbf{U}[\mathbf{index}])),$$

then we set $\mathbf{B}[\mathbf{index}] := \mathbf{false}$.

Step 4. We select the following set of grids:

$$Q_{\mathcal{COM}} = \left\{ \mathbf{U}[\mathbf{index}] \times \mathbf{index} \mid \mathbf{index} \in \mathcal{P}(Q_\rho) \ \& \ \mathbf{B}[\mathbf{index}] \right\}.$$

Step 5. $\delta_{\mathcal{COM}}$, $S_{\mathcal{COM}}$ and $F_{\mathcal{COM}}$ are defined by definition of automaton $\mathcal{COM}(L)$ given before. \square

Both these algorithms have evident simplified modifications. For obtaining them, let us consider the following directed graph of subsets of the set Q_π :

- for each element $\tilde{Q} \subseteq Q_\pi$ except \emptyset , we have a vertex labeled by \tilde{Q} ; this label symbolizes the union of corresponding elements of Q_π ;
- we have the edge from \tilde{Q}' to \tilde{Q}'' (we shall write $\tilde{Q}' \xrightarrow{\mathcal{DG}} \tilde{Q}''$) if and only if for some $A \in Q_\pi$, we have $\tilde{Q}' = \tilde{Q}'' \cup \{A\}$.

Let us denote this directed graph by $\mathcal{DG}(Q_\pi)$. For each its vertex $\tilde{Q} \in \mathcal{P}(Q_\pi)$, let us define its level by

$$|Q_\pi| - |\tilde{Q}|;$$

for example, vertex Q_π has level 0, and for each vertex $A \in Q_\pi$, vertex $\{A\}$ has level $|Q_\pi| - 1$. Thus, by definitions of grids we obtain the following simplification of Algorithm 2.

Algorithm 3 (Constructing automaton \mathcal{U}_1)

Step 1. Consider Boolean array $B[\text{index}]$, where index can be each element of $\mathcal{P}(Q_\pi)$ (except \emptyset). For each possible index , we set

$$B[\text{index}] := (\exists X \in Q_\rho) \left(\text{index} = \left\{ A \in Q_\pi \mid A \# X \right\} \right).$$

Step 2.

for $i:=1$ to $|Q_\pi| - 1$ do

 for each vertex of level i (let this vertex be index)

 execute following Step 3

Step 3.

 if for there exist 2 (or more) vertices ind of level $i-1$,
 such that condition $B[\text{ind}]$ holds and $\text{ind} \xrightarrow{\mathcal{DG}} \text{index}$

 then $B[\text{index}] := \text{true}$

Step 4. We select the following set of grids:

$$Q_{\mathcal{COM}} = \left\{ \text{index} \times \bigcap_{A \in \text{index}} \left\{ X \in Q_\rho \mid A \# X \right\} \mid \text{index} \in \mathcal{P}(Q_\pi) \ \& \ B[\text{index}] \right\}.$$

Step 5. $\delta_{\mathcal{COM}}$, $S_{\mathcal{COM}}$ and $F_{\mathcal{COM}}$ are defined by definition of automaton $\mathcal{COM}(L)$ given before.

Let us remark, that considering subsets of Q_π and Q_ρ (as indexes of arrays), we could consider also element \emptyset ; in this case, \emptyset would correspond to the possible “dead” state of the equivalent canonical automaton.

For Algorithm 3, we shall not formulate its “mirror image”.

6 The detailed example

Let us continue to consider the example of language of [6, Section 3]. For it, let us depict once again automata \tilde{L} and \tilde{L}^R for that language (Figure 1 and 2):

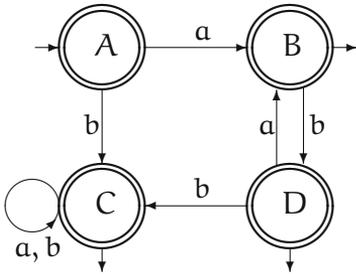


Figure 1

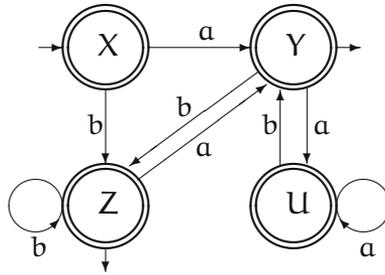


Figure 2

and also its binary relation (Table 1):

#	X	Y	Z	U
A	-	#	#	-
B	#	-	#	-
C	#	#	#	#
D	#	#	#	-

Table 1

Let us remark, that in [6, Section 3] we simply *indicate* the set of grids; and in this paper, we *use the algorithm* of their constructing. Thus, consider using Algorithm 3.

The directed graph \mathcal{DG} for all nonempty subsets of Q_π is given in Figure 3 (the subsets are marked here simply by the strings consisting of their elements). For this figure, we have the following comments. Sets marked by 3 ovals (i.e.,

$\{A, B, C, D\}$, $\{A, C, D\}$, $\{B, C, D\}$ and $\{D\}$) were selected by Step 1 of Algorithm 3.

Using Steps 2 and 3, we consider other subsets (i.e., vertices of graph \mathcal{DG}). Considering them, we have the only vertex (i.e., $\{C, D\}$), for which there exist at least 2 vertices, such that we have edges from them in $\{C, D\}$; we marked this “new suitable” vertex by many ovals. Thus, all the 5 mentioned vertices (and only they) are elements $\alpha(\mathcal{B})$ for some grid \mathcal{B} .

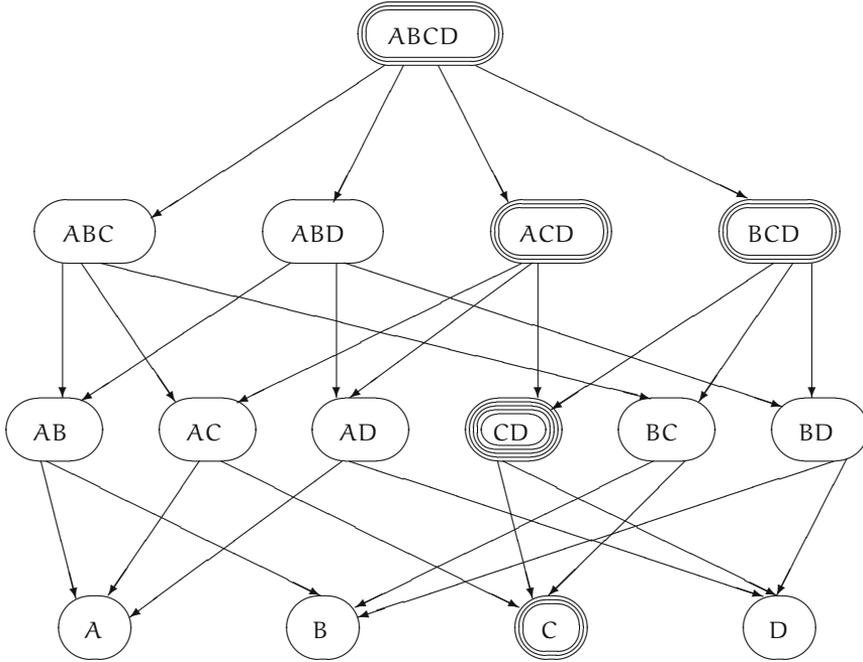


Figure 3

By Step 4 of Algorithm 3, we select for them the following set of grids:⁴

$$\begin{aligned} \zeta &= \{A, C, D\} \times \{Y, Z\}, \quad \eta = \{A, B, C, D\} \times \{Z\}, \quad \vartheta = \{B, C, D\} \times \{X, Z\}, \\ \nu &= \{C\} \times \{X, Y, Z, U\}, \quad \xi = \{C, D\} \times \{X, Y, Z\}. \end{aligned}$$

These letters (i.e., ζ , η , ϑ , ν and ξ) will symbolize states of automaton \mathcal{U}_L .

⁴ Remark once again, that in [6] we simply indicate the set of grids. And in this section, we used Algorithm 3 for their construction.

By Step 5 of Algorithm 3, we simply obtain the following sets of inputs and outputs of \mathcal{U}_1 :

$$S_{\omega\mathcal{M}} = \{\zeta, \eta\}, \quad F_{\omega\mathcal{M}} = \{\vartheta, \nu, \xi\}.$$

And also by Step 5 of Algorithm 3 (i.e., by definition of automaton $\mathcal{COM}(L)$ of Section 3), we obtain transition function $\delta_{\omega\mathcal{M}}$ in the following way.

First at all, consider the following Table 2:

grids (\mathcal{B})	$\alpha(\mathcal{B})$	$\xrightarrow[\delta_\pi]{a}$	ζ	η	ϑ	ν	ξ
ζ	ACD	BC	-	+	+	-	-
η	ABCD	-	-	-	-	-	-
ϑ	BCD	-	-	-	-	-	-
ν	C	C	+	+	+	+	+
ξ	CD	BC	-	+	+	-	-

Table 2

We use here simplified notation as we did before. I.e., elements of the 2nd and the 3rd columns of this table are the sets of elements of Q_π . However, we have to explain this notation detailed, because the symbol “-” in the 3rd column does *not* symbolize the empty set.

Elements of the 2nd column are elements $\alpha(\mathcal{B})$ for the considered grid \mathcal{B} (which is in the 1-st column). If *for each* element of this subset (let this state be A) there exists the transition $A \xrightarrow[\delta_\pi]{a} B$ for some $B \in Q_\pi$, then in this line, the corresponding set of the 3rd column is the union of all such B .⁵ Otherwise, i.e., if for some element A (of the 2nd column) there exists *no* transition $A \xrightarrow[\delta_\pi]{a} B$ for some $B \in Q_\pi$, then in the 3rd column, we set the symbol “-”.

The right part of this table (i.e., since the 4th column) is a square matrix. For each its element in the line marked ζ and in the column marked η , we set + if and only if:

- the set of 3rd column of the line ζ is not “-”;
- and, besides, is a subset of the set of the 3rd column of the line η ;

otherwise we set “-”.⁶ Thus, the right part of the table forms the square

⁵ Let us remind, that we consider the letter a . Remind also, that we consider canonical automata *without* “dead” states.

⁶ Let us especially remark, that the symbol “-” in the 3rd column implies “-” in each cell of this line in the right part of the table.

matrix (i.e., matrix

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix} \quad (5)$$

in the considered example), which formulates, in fact, the condition (1) for the letter \mathfrak{a} .

And for this letter \mathfrak{a} and automaton $\widetilde{\mathbb{L}}^{\mathfrak{R}}$, we obtain the following Table 3:

grids (\mathcal{B})	$\beta(\mathcal{B})$	$\xrightarrow{\frac{\mathfrak{a}}{\delta_\rho}}$	ζ	η	ϑ	ν	ξ
ζ	YZ	$\Upsilon\mathbb{U}$	-	-	-	+	-
η	Z	Y	+	-	-	+	+
ϑ	XZ	Y	+	-	-	+	+
ν	XYZU	$\Upsilon\mathbb{U}$	-	-	-	+	-
ξ	XYZ	$\Upsilon\mathbb{U}$	-	-	-	+	-

Table 3

Similarly, the right part of this table formulates, in fact, the condition (2) for the same letter \mathfrak{a} .

Then the elementwise conjunction of the first matrix (in the considered example, that is matrix (5)) and the transposed matrix of Table 3 gives the matrix for the existence of \mathfrak{a} -transitions. Let us remark, that in the considered examples two matrices for elementwise conjunction are the same;⁷ however, there exist examples where these matrices are different.

Let us consider such tables for letter \mathfrak{b} . For automaton $\widetilde{\mathbb{L}}$, we obtain the following Table 4:

grids (\mathcal{B})	$\alpha(\mathcal{B})$	$\xrightarrow{\frac{\mathfrak{b}}{\delta_\pi}}$	ζ	η	ϑ	ν	ξ
ζ	ACD	C	+	+	+	+	+
η	ABCD	CD	+	+	+	-	+
ϑ	BCD	CD	+	+	+	-	+
ν	C	C	+	+	+	+	+
ξ	CD	C	+	+	+	+	+

Table 4

And for automaton $\widetilde{\mathbb{L}}^{\mathfrak{R}}$, we obtain the following Table 5:

⁷ For the letter \mathfrak{b} , two corresponding matrices also are the same. See below.

grids (\mathcal{B})	$\beta(\mathcal{B})$	$\xrightarrow[\delta_p]{a}$	ζ	η	ϑ	ν	ξ
ζ	YZ	Z	+	+	+	+	+
η	Z	Z	+	+	+	+	+
ϑ	XZ	Z	+	+	+	+	+
ν	XYZU	YZ	+	-	-	+	+
ξ	XYZ	Z	+	+	+	+	+

Table 5

Therefore, we obtain the following matrix of **b**-transitions for automaton $\mathcal{COM}(L)$:

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} \tag{6}$$

(the order of the grids is previous, i.e., $\zeta, \eta, \vartheta, \nu, \xi$).

Using matrices (5) and (6), we simply obtain the following automaton $\mathcal{COM}(L)$ for the considered language:

	$\mathcal{COM}(L)$	a	b
\rightarrow	ζ	η, ϑ	$\zeta, \eta, \vartheta, \nu, \xi$
\rightarrow	η	-	$\zeta, \eta, \vartheta, \xi$
\leftarrow	ϑ	-	$\zeta, \eta, \vartheta, \xi$
\leftarrow	ν	$\zeta, \eta, \vartheta, \nu, \xi$	$\zeta, \eta, \vartheta, \nu, \xi$
\leftarrow	ξ	η, ϑ	$\zeta, \eta, \vartheta, \nu, \xi$

Table 6

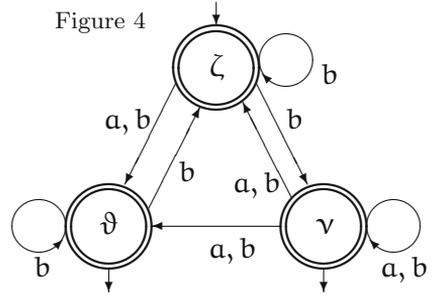
For this automaton, let us also consider its covering subset of grids. One of them⁸ is the following one: $\{\zeta, \vartheta, \nu\}$. And using definition of the covering automaton (Section 3), we obtain the following *corresponding* covering automaton $\mathcal{COM}_{\{\zeta, \vartheta, \nu\}}(L)$ (i.e., the covering automaton for the set $\{\zeta, \vartheta, \nu\}$, see Table 7 and Figure 4). It is easy to prove, that the last automaton does define the given language.

Thus, the considered example gives the *equivalent* covering automaton. However, there are examples of languages, where there exist covering automata which does *not* define the given languages. We shall continue to consider such examples in our following papers.

⁸ As we said before, we shall not consider algorithms of constructing such subsets for arbitrary automaton. For our example there is evidently, that there exist *the only* covering subset containing no more than 3 grids.

$\mathcal{COM}_{\{\zeta, \vartheta, \nu\}}(L)$	a	b
\rightarrow	ζ	ζ, ϑ, ν
\leftarrow	ϑ	ζ, ϑ
\leftarrow	ν	ζ, ϑ, ν

Table 7



7 A series of examples

In [4, Th. 5.1], tight upper bound $D(k)$ on the size of the universal automaton was obtained. $D(k)$ happens to be the k th Dedekind number ([2] etc.), where k is the number of states of automaton accepting given language L .

In this section we consider examples which shows how fast a number of grids (i.e. the size of the universal automaton) can grow if we are given the sizes of two *canonical* automata (i.e., \tilde{L} and \tilde{L}^R). Authors think that these examples supplement the examples considered in [4].

First of all, let us consider the following matrix of dimension 12×12 :

$$\begin{pmatrix}
 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\
 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\
 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\
 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\
 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1
 \end{pmatrix} \tag{7}$$

Evidently, we can consider such matrices for each $x \geq 2$ ($x \times x$ is the dimensions of the “block of zeros”) and each $n \geq 2$, such that $n \bmod x = 0$ ($n \times n$ is the dimensions of the matrix). In the above example (7) we have $x = 4$ and $n = 12$. (Moreover, we can take blocks of zeros of different sizes, see below). We shall not write the strict formulas for the elements of such matrices.

For each matrix of this type we can consider corresponding automaton having the same table of $\#$ -relation. And by [7], such an automaton always exists.

Constructing *some* grids for the example (7), we can select exactly one line of $\{1, 2, 3, 4\}$, then exactly one line of $\{5, 6, 7, 8\}$, and also exactly one line of $\{9, 10, 11, 12\}$; obviously, the numbers of columns must be the same as the numbers of lines. Thus, in the example (7) we have at least $4^3 = 64$ grids.

Next, let

$$\mathbf{n} = \mathbf{a}_1 + \mathbf{a}_2 + \dots + \mathbf{a}_k,$$

and we have blocks of zeros of the sizes $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k$; then corresponding number of blocks (states of canonical automaton) will be no less than

$$\mathbf{a}_1 \cdot \mathbf{a}_2 \cdot \dots \cdot \mathbf{a}_k$$

. Our task is to find the maximum value of this expression with fixed \mathbf{n} . To do so, we should take into account that

$$4 = 2 + 2, \quad 2 \cdot 2 = 4 \geq 4;$$

$$5 = 2 + 3, \quad 2 \cdot 3 = 6 > 5;$$

$$6 = 2 + 2 + 2 = 3 + 3, \quad 3 \cdot 3 > 2 \cdot 2 \cdot 2 = 6 \quad \text{etc.}$$

i.e. each summand, greater or equal 4, can be split into 2's and 3's, while keeping the value of product at least as big. It is also obvious, that splitting the summands we should prefer 3's (see the last inequality).

So, the sought-for maximum has one of the following types:

$$3 \cdot 3 \cdot \dots \cdot 3$$

$$3 \cdot 3 \cdot \dots \cdot 3 \cdot 2 \cdot 2$$

$$3 \cdot 3 \cdot \dots \cdot 3 \cdot 2,$$

depending on the \mathbf{n} modulo 3. For simplicity's sake we shall limit ourselves with the case $\mathbf{n} \bmod 3 = 0$.

To sum up, our examples shows that the size of the universal automaton can grow exponentially (with base $3^{1/3}$) with regard to the size of canonical automata, i.e. with regard to $\mathbf{n} = \min(|Q_\pi|, |Q_\rho|)$. Let us repeat that our examples *do not* give the exact number of blocks which may form if we are given only the value of $\mathbf{n} = \min(|Q_\pi|, |Q_\rho|)$ for automata $\widetilde{\mathbf{L}}$ and $\widetilde{\mathbf{L}}^{\mathbf{R}}$.

Note that we must not combine this examples with an obvious bound $\mathbf{n} = 2^k$ for the number of states of canonical automata, since the resulting function $3^{\mathbf{n}/3} = 3^{2^k/3}$ grows much faster than $D(k)$. This only means that automata with $\#$ -relation similar to (7) cannot have equivalent automata with number of states much less then \mathbf{n} .

8 Conclusion

In the next paper we are going to consider:

- the loops of the basis automaton $\mathcal{BA}(L)$ and of automaton $\mathcal{COM}(L)$;
- the consideration of the covering automaton, does *not* define the given language (unlike Section 6); i.e., in fact, the consideration of automaton Waterloo ([3]) from the point of view of the basis automaton;
- the *constructive* proof of the following fact: examples like Waterloo can be constructed *for each* table of relation $\#$ having the following additional property: there exists the *proper* covering subset.

9 Acknowledgement

Authors would like to express the gratitude to Professor Anton Klyachko (Moscow State University) for making valuable comments.

References

- [1] A. Aho, J. Ullman, *The Theory of Parsing, Translation, and Compiling, Vol. 1, Parsing*. Prentice Hall, 1972. $\Rightarrow 8$
- [2] R. Dedekind, *Über Zerlegungen von Zahlen durch ihre größten gemeinsamen Teiler*. Gesammelte Werke, Vol. 2, 103–148. $\Rightarrow 18$
- [3] T. Kameda, P. Weiner, On the state minimization of nondeterministic finite automata, *IEEE Trans. on Comp.* **C-19**, 7 (1970) 617–627. $\Rightarrow 20$
- [4] S. Lombardy, J. Sakarovitch, The Universal Automaton, in: *Logic and Automata, Texts in Logic and Games* Amsterdam Univ. Press. Vol. 2 (2008) 457–504. $\Rightarrow 5, 7, 18$
- [5] B. Melnikov, Extended nondeterministic finite automata, *Fundamenta Informaticae* **104**, 3 (2010) 255–265. $\Rightarrow 5, 6$
- [6] B. Melnikov, Once more on the edge-minimization of nondeterministic finite automata and the connected problems, *Fundamenta Informaticae* **104**, 3 (2010) 267–283. $\Rightarrow 5, 6, 9, 10, 13, 14$
- [7] B. Melnikov, A. Melnikova, Some more on the basis finite automaton, *Acta Univ. Sapientiae, Inform.* **5**, 2 (2013) 227–244 $\Rightarrow 19$
- [8] B. Melnikov, N. Sciarini-Guryanova, Possible edges of a finite automaton defining a given regular language, *The Korean Journal of Computational and Applied Mathematics* **9**, 2 (2002) 475–485. $\Rightarrow 5$

Received: June 9, 2013 • Revised: August 2, 2013