Artificial Intelligence and Human-Computer Interaction
Y. Ye and P. Siarry (Eds.)

doi:10.3233/FAIA240142

On the Knowledge-Based Study of All Semilattices of Waterloo Automaton

Mikhail ABRAMYAN [a,b,1] and Boris MELNIKOV [a]
[a] *Shenzhen MSU–BIT University, Shenzhen, China*
[b] *Southern Federal University, Rostov-on-Don, Russian Federation*
ORCiD ID: Mikhail Abramyan https://orcid.org/0000-0002-2802-6144
ORCiD ID: Boris Melnikov https://orcid.org/0000-0002-6765-6800

Abstract. The aim is to study the set of subsets of grids of Waterloo automaton and the set of covering automata defined by the grid subsets. The study was carried out using the library for working with nondeterministic finite automata NFALib implemented by one of authors (M. Abramyan) in C#. The results are regularities obtained when considering semilattices of covering automata for Waterloo automaton. A complete description of the obtained semilattices from the point of view of equivalence of the covering automata to the original Waterloo automaton is given, the criterion of equivalence of the covering automaton to Waterloo automaton in terms of properties of the subset of grids defining the covering automaton is formulated. The relevance of the subject area under consideration is due to the need to research of a set of regular languages and, in particular, description of their various subclasses. Also relevant are the problems that may arise in some subclasses. This will give, among other things, the possibility of describing new algorithms for the equivalent transformation of nondeterministic finite automata.

Keywords. Nondeterministic Finite Automata, universal automaton, grid, covering automaton, equivalent transformation algorithms, Waterloo automaton

1. Introduction

The paper continues the study of semilattices arising in the analysis of regular languages and related finite automata. The papers [1-3] are devoted to similar problems.

There are different complete invariants for describing a regular language: not only well-known canonical automata [4-6], but also basis automata [7] and universal automata [8]. When constructing basis automata and universal automata, we need to construct canonical automata both for a given regular language and for its mirror image. In the process of such a construction, we can obtain, among other objects, a special binary relation # defined on the state pairs of these two canonical automata. This relation is also invariant (but incomplete) for the language under consideration.

The Waterloo language and the associated Waterloo automaton are currently the most interesting for the study. The universal automaton constructed for this language has the following property: among its covering automata [8-9], there exists a non-

[1] Corresponding Author: Mikhail Abramyan, Shenzhen MSU–BIT University, Shenzhen, China. E-mail: m-abramyan@yandex.ru.

equivalent one (see Section 2). This fact is directly related to the development of efficient state minimization algorithms for nondeterministic finite automata.

For any automaton, the set of covering automata associated with it forms a semilattice by union, but it can be shown that, in general, it does not form a semilattice by intersection (see Section 3). More concretely, instead of a single semilattice by intersection, it forms a union of several such semilattices. The present paper is devoted to the consideration of such a construction for Waterloo automaton and continues the study of Waterloo automaton semilattices begun in [10-11].

Section 2 of the paper contains preliminary information; terminology related to finite automata is given, mainly those terms related to a universal automaton and to grids, which can be considered as states of this automaton. All the concepts are illustrated on the example of Waterloo automaton; at the same time, a general approach to the program study of automata for Waterloo-like languages is demonstrated using the NFALib library implemented by M. Abramyan in C# [12].

Section 3 discusses the properties of semilattices of covering automata and gives general properties of semilattices for Waterloo automaton. Sections 4 and 5 contain a more detailed study of various semilattices, connected, first of all, with the presence of covering automata which are not equivalent to the original Waterloo automaton. Section 6 gives and justifies the criterion of equivalence of the covering automaton to Waterloo automaton in terms of grids defining the covering automaton.

2. Basic concepts and related examples forWaterloo automaton

In this section, we give the necessary notions related to finite automata and illustrate them on the example of Waterloo automaton W (Figure 1). The interest in this automaton is due, in particular, to the fact that it allows us to identify important features of the state minimization algorithm of nondeterministic finite automata. Waterloo automaton was first described in [13]. It is the objects related to Waterloo automaton that will be the main subject of consideration in the following sections.

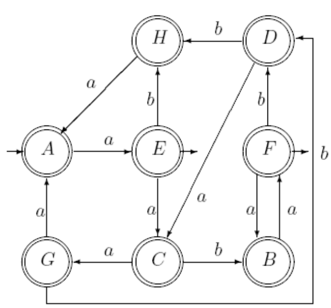

Figure 1. Waterloo automaton W.

Since the paper describes the results of the program study of automata based on the application of the NFALib library, when describing each object, the library method that allows to obtain this object is specified.

The analyzed automaton K (defining some regular language L) is created as an object of `NFA` type on the basis of information from a text file. Figure 2 shows the contents of the file corresponding to Waterloo automaton. The meaning of the notations should be clear from comparing this figure with Figure 1.

```
NFA w = NFA.FromFile("Waterloo.txt");
```

```
%% Waterloo
         a   b
==> A    E   -
    B    F   -
    C    G   B
    D    C   H
<== E    C   H
<== F    B   D
    G    A   D
    H    A   -
```

Figure 2. Text description of the automaton W.

The state minimization algorithm is based on the analysis of a *binary relation* [6, Sec. 3.3] connected with the sets of states X and Y of two canonical automata, which are based on the analyzed nondeterministic finite automaton K and its mirror automaton K^R. Waterloo automaton is a deterministic automaton that does not change after its canonicalization. The canonical automaton for the mirror automaton takes the form given in Figure 3. To obtain mirror and canonical automata, the NFALib library provides the GetMirror and GetCanonicalDFA methods and CurrentCanonicalDFA and CurrentMirrorCanonicalDFA properties.

```
NFA w2 = w.CurrentMirrorCanonicalDFA
  .GetRenamedStates(i => names[i]);
```

```
         a   b   % States
==> X    Y   -   % E-F
<== Y    Z   U   % A-B
    Z    V   W   % F-G-H
    U    W   -   % C
    V    P   U   % B-C
    W    Q   R   % D-E
    P    Y   R   % D-E-F
<== Q    S   -   % A
    R    V   -   % F-G
    S    U   W   % G-H
```

Figure 3. Canonical automaton for the automaton W^R.

The States column in Figure 3 shows the sets of states before their renaming; each such state corresponds to the set of states of the original mirror automaton obtained as a result of determinization of this automaton.

On the basis of the canonical automata, we construct a *matrix of relation* #. Rows of this matrix correspond to the states of the canonical automaton for the automaton K, and the columns correspond to the states of the canonical automaton for the mirror automaton K^R. A view of the matrix of relation # for the automaton W is given in Figure 4. To get the matrix of relation # of some automaton in the NFALib library, it is enough to refer to its CurrentSharpRelation property.

The elements of the matrix labeled with # are defined as follows: in the row corresponding to some state s of the canonical automaton for the original automaton (in our case, for the automaton W), the elements for those states of the canonical automaton for the mirror automaton that contain the state s in the column States are labeled. For example, for the row A, the columns Y and Q are marked, because the state Y corresponds to the set of states $\{A, B\}$, and the state Q corresponds to the set of states $\{A\}$ (see Figure 3).

```
SharpRelation sharpRelation = w.CurrentSharpRelation;
```

```
%% NFA: Waterloo
  X Y Z U V W P Q R S
A - # - - - - - # - -
B - # - - # - - - - -
C - - - # # - - - - -
D - - - - - # # - - -
E # - - - - # # - - -
F # - # - - - # - # -
G - - # - - - - - # #
H - - # - - - - - - #
```

Figure 4. Matrix of relation # for automaton W.

A set of *grids* is associated with the relation # [6, Section 3.4]. Each grid is defined by a pair of subsets $X_0 \in X$ and $Y_0 \in Y$, where X is the set of rows of the matrix of relation # (coinciding with the set of states of the canonical automaton for the automaton K), and Y is the set of columns of the matrix of relation # (coinciding with the set of states of the canonical automaton for the automaton K^R). The subsets X_0 and Y_0 must satisfy two conditions: (1) for any states $x \in X_0$ and $y \in Y_0$, the relation $x \# y$ is satisfied; (2) the subsets X_0 and Y_0 cannot be expanded while preserving condition (1). We denote such a grid by $X_0 \times Y_0$.

A set M of grids is called a *covering set* if for any elements $x \in X$, $y \in Y$ such that $x \# y$, there exists a grid $X_0 \times Y_0$ of M for which $x \in X_0$ and $y \in Y_0$. Clearly, the complete set of grids constructed by the relation # is a covering set. In Figure 5, we give the complete set of 14 grids for the relation # corresponding to the automaton W. To get the complete set of grids, the NFALib library provides the GetCompleteGrids method based on the brute force method.

```
List<Grid> completeGrids =
  sharpRelation.GetCompleteGrids().ToList();
```

```
Complete Grids
{ A } × { Y, Q }  % 1
{ A, B } × { Y }  % 2
{ B } × { Y, V }  % 3
{ B, C } × { V }  % 4
{ C } × { U, V }  % 5
{ D, E } × { W, P }  % 6
{ E } × { X, W, P }  % 7
{ E, F } × { X, P }  % 8
{ F } × { X, Z, P, R }  % 9
{ F, G } × { Z, R }  % 10
{ G } × { Z, R, S }  % 11
{ G, H } × { Z, S }  % 12
{ F, G, H } × { Z }  % 13
{ D, E, F } × { P }  % 14
```

Figure 5. Complete set of grids for the automaton W.

In [8], an algorithm is described which allows to construct a *universal automaton* $COM(K)$, which defines the same regular language L as the original automaton K, and each grid corresponds to some state of the automaton $COM(K)$. The automaton $COM(W)$ is shown in Figure 6. The method GetCOM(completeGrids) of the NFALib library is intended for obtaining it.

The constructed automaton *COM*(*W*) is equivalent to the original automaton *W*, which can be shown by canonization it using the GetCanonicalDFA method and appropriate renaming its states.

```
NFA com = w.GetCOM(completeGrids, "COM");
```

```
%% COM
        a              b         % Grids
==> 1   6,7,8,14       -         % { A } × { Y, Q }
==> 2   8,14           -         % { A, B } × { Y }
    3   8,9,10,13,14   -         % { B } × { Y, V }
    4   10,13          -         % { B, C } × { V }
    5   10,11,12,13    2,3,4     % { C } × { U, V }
    6   4,5            12,13     % { D, E } × { W, P }
<== 7   4,5            12,13     % { E } × { X, W, P }
<== 8   4              -         % { E, F } × { X, P }
<== 9   2,3,4          6,14      % { F } × { X, Z, P, R }
    10  2              6,14      % { F, G } × { Z, R }
    11  1,2            6,14      % { G } × { Z, R, S }
    12  1,2            -         % { G, H } × { Z, S }
    13  2              -         % { F, G, H } × { Z }
    14  4              -         % { D, E, F } × { P }
```

Figure 6. Automaton *COM*(*W*).

Based on the *COM*(*K*) automaton, one can define a family of *covering automata*, each of which is obtained by removing some states of the *COM*(*K*) automaton, with the remaining states corresponding to the grids forming the covering set.

The algorithm for minimizing the original automaton *K* consists in choosing a covering set of grids M_0 of minimal size for which the covering automaton built on its basis is equivalent to the automaton *K*, i. e., defines the same regular language *L*.

Unfortunately, *not every covering set of grids yields a covering automaton equivalent to the original one*. An example is Waterloo automaton. The minimal covering set for it is the set M_0 of 7 elements containing the following grids from the complete set: 1, 3, 5, 6, 8, 10, 12 (the GetMinGridCovers method is provided for finding minimal covering sets).

Figure 7 shows the covering automaton W_0 constructed from the minimal covering set M_0 using the GetCovering method applied to the automaton *COM*(*W*).

```
NFA w4 = com.GetCovering(minGrids);
```

```
        a       b     % Grids
==> 1   6,8     -     % { A } × { Y, Q }
    3   8,10    -     % { B } × { Y, V }
    5   10,12   3     % { C } × { U, V }
    6   5       12    % { D, E } × { W, P }
<== 8   -       -     % { E, F } × { X, P }
    10  -       6     % { F, G } × { Z, R }
    12  1       -     % { G, H } × { Z, S }
```

Figure 7. Minimal covering automaton W_0 for the automaton *COM*(*W*).

After the procedure of canonization and renaming of states, the automaton W_0 takes the form shown in Figure 8. This automaton is not equivalent to the original

Waterloo automaton (compare with Figure 2; the different line corresponds to the state *F*).

```
         a   b   % States
==> A    E   -   % 1
    B    F   -   % 3
    C    G   B   % 5
    D    C   H   % 6
<== E    C   H   % 6-8
<== F    -   D   % 8-10
    G    A   D   % 10-12
    H    A   -   % 12
```

Figure 8. Canonical automaton for a minimal covering automaton with renamed states.

3. Semilattices of covering automata

Since the minimal covering automaton, as the example of Waterloo automaton shows, will not necessarily be equivalent to the original automaton, it is of interest to investigate the entire set of covering automata that can be derived from the *COM* automaton for different covering sets of grids.

Note that all covering sets of grids, according to the definitions given in the previous section, form a *semilattice by union*. This is obvious since the union of any two covering sets is also a covering set. In this sense, we can also speak of a semilattice by union for all covering automata, understanding by the union operation the union of subsets of grids on the basis of which these covering automata are constructed.

However, the complete set of covering sets of grids does not form a semi-lattice by intersection. To prove this fact, it is enough to note that there is a covering set of grids {1, 2, 4, 5, 6, 8, 10, 12} whose intersection with the minimal covering set M_0 gives a set {1, 5, 6, 8, 10, 12} which is not a covering set (since its size is smaller than the size of the minimal covering set M_0).

Although the complete set of covering sets of grids does not, in general, form a semilattice by intersection, it is possible to obtain subsets of covering sets which do form such a semilattice. Each such *semilattice by intersection* will include some *base covering set* of grids, the complete set, and all intermediate sets, with the property that any pairwise intersection of them also belongs to this semilattice. We will consider only such semilattices by intersection that cannot be extended. It should be noted that each such semilattice can be interpreted as a *hypercube* of the corresponding dimension.

In what follows, we will only consider semilattices by intersection that cannot be extended.

For most automata, it is not of interest to consider sets of covering automata and associated semilattices, since all these automata are equivalent. A different situation arises in the study of Waterloo automaton *W*. As it was shown in the previous section, for the matrix of relation # associated with this automaton, 14 grids can be constructed, from which we can select a single minimal covering set M_0 of size 7. The covering automaton W_0 constructed on the basis of the set M_0 is not equivalent to the original automaton *W*.

The study of the grids of the matrix of relation # for Waterloo automaton shows that there are 260 different covering sets of grids, the sizes of which vary from 7 to 14. The distribution of sizes of covering sets is given in Table 1.

Based on different covering sets of grids, 8 different semilattices can be constructed (recall that we consider semilattices by intersection that cannot be extended). The characteristics of these semilattices are given in Table 2.

Table 1. Distribution of covering sets by the number of grids for automaton *W*.

Number of grids in the covering set	7	8	9	10	11	12	13	14
Number of covering sets	1	11	43	79	76	39	10	1

Table 2. Semilattices for Waterloo automaton.

Semilattice	Base covering set of grids	Number of elements (covering automata)	Number of covering automata not equivalent to *W*
A	{1, 3, 5, 6, 8, 10, 12}	128	48
B	{1, 2, 4, 5, 6, 8, 10, 12}	64	0
C	{1, 3, 5, 6, 7, 9, 10, 12}	64	0
D	{1, 3, 5, 6, 7, 9, 11, 12}	64	0
E	{1, 3, 5, 6, 8, 9, 11, 12}	64	0
F	{1, 2, 4, 5, 6, 7, 9, 10, 12}	32	8
G	{1, 2, 4, 5, 6, 7, 9, 11, 12}	32	12
H	{1, 2, 4, 5, 6, 8, 9, 11, 12}	32	8

Of particular interest is the last column of Table 2, which shows the number of covering automata from a given semilattice that are not equivalent to the original Waterloo automaton. In the following sections, we will discuss the peculiarities of different semilattices related to the presence of non-equivalent covering automata in them.

4. Study of the maximal semilattice for Waterloo automaton

This section describes additional properties of the semilattice *A* from Table 2. This semilattice, first, has maximal size (128 elements), second, its basic element is a covering automaton W_0 constructed by the minimal covering set of grids and, third, it contains the largest number of covering automata which are not equivalent to Waterloo automaton.

Further investigation of the semilattice *A* shows that it contains four sets of pairwise equivalent automata, including three sets of automata that are not equivalent to Waterloo automaton:

- the set N_1 that contains the covering automaton corresponding to the set $M_0 = \{1, 3, 5, 6, 8, 10, 12\}$,
- the set N_2 that contains the covering automaton corresponding to the set $M_1 = \{1, 2, 3, 5, 6, 8, 10, 12\}$,
- the set N_3 that contains the covering automaton corresponding to the set $M_2 = \{1, 3, 4, 5, 6, 8, 10, 12\}$.

Each of them contains 16 elements forming 4-dimensional hypercube. The set of covering automata equivalent to Waterloo automaton will be denoted by N_0.

Each of the eight 4-dimensional hypercubes into which the original semilattice (7-dimensional hypercube) decomposes is characterized by a special combination of three additional grids with numbers 2, 4, and 9. Namely, if the basic covering set $M_0 = \{1, 3,$

5, 6, 8, 10, 12}, complemented by grids from the additional *ordered*[2] set D_0 = (2, 4, 7, 9, 11, 13, 14), does not include any of grids 2, 4, 9, we obtain the covering automaton from set N_1; if only grid 2 is included, we obtain the covering automaton from set N_2, if only grid 4 is included, we obtain the covering automaton from set N_3. The presence of grid 9 or the simultaneous presence of grids 2 and 4 guarantees that the resulting covering automaton is equivalent to the original Waterloo automaton and thus belongs to N_0.

These relations are presented in Table 3. Grid combinations are specified by a 7-character string, which defines the presence or absence of each of the grids included in D_0 = (2, 4, 7, 9, 11, 13, 14) as follows: if some grid from D_0 is *included* in all elements of the set, then its position in the string is indicated by 1, if the grid is *not included*, then 0 is indicated, if the grid can be both included and not included in the set, then an asterisk * is indicated at its position. For example, the notation 01*0*** for the set N_2 means that this set includes covering sets of grids that include grid 4 and do *not* include grids 2 and 9 (grids 7, 11, 13, 14 may or may not be included in the covering set).

Table 3. Subsets of the semilattice A.

A set that includes the elements of a subset	Description of the subset (presence or absence of grids 2, 4, 7, 9, 11, 13, 14)	Subset size	Are the elements equivalent to the automaton *W*?
N_1 (the whole set)	00*0***	16	no
N_2 (the whole set)	01*0***	16	no
N_3 (the whole set)	10*0***	16	no
N_0	11*0***	16	yes
N_0	***1***	64	yes

Thus, a necessary and sufficient condition for a covering automaton from a semilattice A to be equivalent to Waterloo automaton is that there exists a grid 9 or simultaneously grids 2 and 4 in the corresponding covering set.

5. Study of other semilattices for Waterloo automaton

Let us turn to the remaining semilattices given in Table 2. They can be divided into two groups. The first group includes semilattices *B*, *C*, *D*, *E*, for which the basic covering automaton is equivalent to the automaton *W*. Therefore, all other elements of these semilattices are equivalent to the automaton *W*, and no additional study of these semilattices is required.

Unlike semilattices *B*, *C*, *D*, *E*, semilattices *F*, *G*, *H* contain covering automata which are not equivalent to the automaton *W*. Additional analysis shows that these automata can be divided into 5 sets: N_4, N_5, N_6, N_7, N_8, each of which contains automata equivalent to each other and not equivalent to automata from other sets (as well as to automata from sets N_0, N_1, N_2, N_3 described in the previous section). Sets N_4 and N_5 are completely contained in semilattice *F*, set N_6 is completely contained in semilattice *G*, and sets N_7 and N_8 are completely contained in semilattice *H*. Moreover, the semilattice *G* includes half of the elements included in each of the sets N_4, N_5, N_7, and N_8.

A more detailed analysis of the semilattices *F*, *G*, *H* and the associated sets of covering automata not equivalent to the automaton *W* is convenient to perform together, since the sets N_4, N_5, N_7, and N_8 are parts of several semilattices.

[2]We fix the order of the elements of the set D0, as this will allow us to easily define its different subsets, as will be described later.

Once again, we show the base covering sets of grids for semilattices F, G, and H:
F: {1, 2, 4, 5, 6, 7, 9, 10, 12},
G: {1, 2, 4, 5, 6, 7, 9, 11, 12},
H: {1, 2, 4, 5, 6, 8, 9, 11, 12}.

Since all these sets contain grids from the set M' = {1, 2, 4, 5, 6, 9, 12}, we can exclude these grids from further consideration and restrict ourselves to analyzing the grids included in the *ordered* set D_1 = (3, 7, 8, 10, 11, 13, 14). For different subsets of the set D_1, we will use notations similar to those previously used for subsets of the set D_0 (see Table 3): a subset is defined by a string of 7 characters; if some grid from D_1 is *included* in all elements of the subset, then its position in the string is indicated by 1, if the grid is *not included*, then 0 is indicated, if the grid can both be included and not included in the elements of the subset, then its position is indicated by an asterisk *. Given such notations, the semilattice F can be described as follows: *1*1***. This means that, in addition to the grids from set M', all its elements include grids 7 and 10, and other grids from D_1 may or may not be included.

The description of semilattices F, G, H is given in Table 4, the description of sets N_4, N_5, N_6, N_7, N_8 is given in Table 5.

Table 4. Semilattices F, G, H.

Semilattice	Description (presence or absence of grids 3, 7, 8, 10, 11, 13, 14)	Size	Contains elements of the following sets
F	*1*1***	32	N_0, N_4 (the whole set), N_5 (the whole set)
G	*1**1**	32	N_0, N_6 (the whole set), N_4, N_5, N_7, N_8
H	**1*1**	32	N_0, N_7 (the whole set), N_8 (the whole set)

Table 5. Sets N_4, N_5, N_6, N_7, N_8.

Set	Description (presence or absence of grids 3, 7, 8, 10, 11, 13, 14)	Size	Included in the following semilattices
N_4	0101**0	4	F (all elements of the set), G (elements with grid 11)
N_5	0101**1	4	F (all elements of the set), G (elements with grid 11)
N_6	01001**	4	G (all elements of the set)
N_7	0*1010*	4	H (all elements of the set), G (elements with grid 7)
N_8	0*1011*	4	H (all elements of the set), G (elements with grid 7)

The analysis of Table 5 shows that *a necessary and sufficient condition* for a covering automaton of semilattices F, G, H to be equivalent to Waterloo automaton is the presence of grid 3 or simultaneously grids 8 and 10 in the corresponding covering set.

6. Criterion of equivalence of covering automata to Waterloo automaton

In the study of the semilattice A in Section 4 and of the semilattices F, G, H in Section 5, a special role of grids 3 and 9, as well as combinations of grids 2, 4 and 8, 10 was revealed. Recall that the presence of grid 9 or simultaneously grids 2 and 4 in the covering set guarantees that the covering automaton included in the semilattice *A* is equivalent to the automaton *W*, and, for the semilattices *F*, *G*, *H*, the similar fact is true if grid 3 or simultaneously grids 8 and 10 are included in the covering set.

These results can be generalized to the whole set of covering automata for Waterloo automaton if we additionally analyze the presence of these special grids in the base covering sets for all semilattices. The corresponding data are summarized in Table 6. In it, for brevity, we denote the *simultaneous* presence of two grids using the "+" operation: 2+4 or 8+10.

Table 6. Additional characteristics of semilattices for Waterloo automaton.

Semilattice	Base covering set of grids	Special grids in the base set	Additional grids providing equivalence to Waterloo automaton
A	{1, 3, 5, 6, 8, 10, 12}	3, 8+10	2+4 or 9
B	{1, 2, 4, 5, 6, 8, 10, 12}	2+4, 8+10	–
C	{1, 3, 5, 6, 7, 9, 10, 12}	3, 9	–
D	{1, 3, 5, 6, 7, 9, 11, 12}	3, 9	–
E	{1, 3, 5, 6, 8, 9, 11, 12}	3, 9	–
F	{1, 2, 4, 5, 6, 7, 9, 10, 12}	2+4, 9	3 or 8+10
G	{1, 2, 4, 5, 6, 7, 9, 11, 12}	2+4, 9	3 or 8+10
H	{1, 2, 4, 5, 6, 8, 9, 11, 12}	2+4, 9	3 or 8+10

It follows from Table 6 that in order for the covering automaton constructed by the *COM*(*W*) automaton to be equivalent to Waterloo automaton *W*, it is necessary and sufficient that the corresponding covering set contains either grids 3 and 9 or grids 2, 4, 8, and 10. If this condition is violated, we obtain covering automata that are not equivalent to Waterloo automaton. There are 8 sets of such automata; each set contains automata equivalent to each other.

The obtained results can be presented in the form of Table7, where the grids from the set {2, 3, 4} that are included in the covering set are indicated vertically, and the grids from the set {8, 9, 10} that are included in the covering set are indicated horizontally. The table has 5 rows and 5 columns, because there are no covering sets that do not contain any grids from sets {2, 3, 4} and {8, 9, 10} or contain only one grid 2 or 4 from set {2, 3, 4} or only one grid 8 or 10 from set {8, 9, 10}. The cells of the table indicate properties of automata with the given set of grids, namely: to which set of pairwise equivalent covering automata they belong and, in parentheses, the number of them. Recall that the covering automata equivalent to Waterloo automaton belong to the set N_0; for clarity, the cells with the set N_0 are additionally marked with the asterisk *.

Table 7. Distribution of sets of equivalent covering automata depending on grids 2, 3, 4, 8, 9, 10 included in the covering sets.

	9	**8, 9**	**8, 10**	**9, 10**	**8, 9, 10**
3	* N_0 (4)	* N_0 (8)	N_1 (16)	* N_0 (8)	* N_0 (16)
2, 3	* N_0 (4)	* N_0 (8)	N_2 (16)	* N_0 (8)	* N_0 (16)
2, 4	N_6 (4)	N_7 (4), N_8 (4)	* N_0 (16)	N_4 (4), N_5 (4)	* N_0 (16)
3, 4	* N_0 (4)	* N_0 (8)	N_3 (16)	* N_0 (8)	* N_0 (16)
2, 3, 4	* N_0 (4)	* N_0 (8)	* N_0 (16)	* N_0 (8)	* N_0 (16)

Acknowledgments

The work is supported by a grant from the scientific program of Chinese universities "Higher Education Stability Support Program" (section "Shenzhen 2022 – Science, Technology and Innovation Commission of Shenzhen Municipality") – 深圳 市 2022 年高等院校定支持划助目.

References

[1] Melnikov BF. Semi-lattices of the subsets of potential roots in the problems of the formal languages theory. Part I. Extracting the root from the language. International Journal of Open Information Technologies. 2022;10(4):1-9. (in Russian).
[2] Melnikov BF. Semi-lattices of the subsets of potential roots in the problems of the formal languages theory. Part II. Constructing an inverse morphism. International Journal of Open Information Technologies. 2022;10(5):1-8. (in Russian).
[3] Melnikov BF. Semi-lattices of the subsets of potential roots in the problems of the formal languages theory. Part III. The condition for the existence of a lattice. International Journal of Open Information Technologies. 2022;10(7):1-9. (in Russian).
[4] Ginsburg S. The mathematical theory of context-free languages. Moscow: Mir Publ. 1973; 301. (in Russian).
[5] Brauer W. Introduction in the finite automata theory. Moscow: Radio I Svyaz Publ. 1987; 390. (in Russian).
[6] Melnikov B. Regular languages and nondeterministic finite automata. Moscow: RGSU Publ. 2018; 179. (in Russian).
[7] Dolgov V, Melnikov B, Melnikova A. The loops of the basis finite automaton and the connected questions. Bulletin of the Voronezh State University. Series: Physics, Mathematics. 2016;4:95-111 (in Russian).
[8] Dolgov V, Melnikov B. Construction of universal finite automaton. II. Examples of algorithms functioning. Bulletin of the Voronezh State University. Series: Physics, Mathematics. 2014;1:78-85 (in Russian).
[9] Melnikov B, Melnikova A. Edge-minimization of non-deterministic finite automata. The Korean Journal of Computational and Applied Mathematics. 2001;8(3):469-479,doi:10.1007/bf02941980.
[10] Zyablitceva LV, KorabelshchikovaSYu, Abramyan ME. Semilattice graphs, matrix representations of idempotent semigroups and Waterloo automaton semilattices. International Journal of Open Information Technologies.2023;11(7):69-76. (in Russian).
[11] Abramyan ME, Melnikov BF. A program study of the union of semilattices on the set of subsets of grids of Waterloo language. Journal of Applied Mathematics and Physics. 2023;11:1459-1470,doi:10.4236/jampp.2023.115095.
[12] Abramyan ME. Computing the weight of subtasks in state minimization of nondeterministic finite automata by the branch and bound method. University proceedings. Volga region. Physical and mathematical sciences. 2021;2(58):46-52(in Russian),doi:10.21685/2072-3040-2021-2-4.
[13] Kameda T, Weiner P. On the state minimization of nondeterministic finite automata. IEEE Transactions on Computers. 1970; C-19(7):617-627,doi:10.1109/t-c.1970.222994.