Journal of Computer and Systems Sciences International, Vol. 44, No. 5, 2005, pp. 800–807.
Translated from Izvestiya Akademii Nauk. Teoriya i Sistemy Upravleniya, No. 5, 2005, pp. 137–145.

ARTIFICIAL INTELLIGENCE

Classification of Structures Generated by One-Dimensional Binary Cellular Automata from a Point Embryo

L. A. Naumov and A. A. Shalyto

St. Petersburg State University of Information Technologies, Mechanics, and Optics, ul. Sablinskaya 14, St. Petersburg, 197101 Russia

Received October 20, 2004

Abstract—Various classifications of structures generated by one-dimensional binary cellular automata from a point embryo are presented. The number of classes of automata is calculated for different invariance operations. It is demonstrated that there exist cellular automata with memory and memoryless cellular automata exhibiting the same behavior.

INTRODUCTION

Cellular automata have been studied in numerous publications, e.g., [1–4]. Of great interest here is the consideration of one-dimensional cellular automata, which generate self-reproducing configurations such as the Sierpinsky gasket and the binary Pascal triangle. In papers [5, 6], these structures are obtained by rather elaborate mathematical methods. One-dimensional cellular automata are the simplest tools for generating structures like these. Here, by a structure, we mean a sequence of states of a one-dimensional grid [7], i.e., a set of configurations of an automaton arranged one under the other.

All 256 possible structures generated by a one-dimensional binary cellular automaton containing a single point embryo at step zero are listed in paper [4]. Some of these structures are equivalent with respect to invariant operations such as mirror reflection, inversion, offset by a single cell, etc.

The purpose of this paper is to classify structures generated by one-dimensional cellular automata from a point embryo with respect to the invariant operations introduced and to calculate the number of representatives in various classifications.

1. SPECIFICATION OF THE NEXT-STATE FUNCTION OF A ONE-DIMENSIONAL BINARY CELLULAR AUTOMATON

We consider cellular automata whose neighborhood pattern consists of the right and left neighbors of a cell (with regard for the toroidal boundary conditions). Since, in this paper, we study the automaton behavior for the number of steps, which is significantly smaller than the grid width, the boundary conditions do not show up.

The next-state function of the automaton, which determines the evolution of the state of a cell is best specified in tabular form. The left column of the table contains the current states of cells $i - 1$, i, and $i + 1$ arranged lexicographically; the right column has the next state of cell i.

This table has eight rows, each containing one of the two possible values of the next state. Thus, there are 256 various next-state functions and, hence, 256 cellular automata for arbitrary initial conditions.

Associate each automaton with the number in the range from 0 to 255 whose binary representation appears in the right column of the next-state table. For example, the function $f(c_{i-1}, c_i, c_{i+1}) = c_i'$ with the transposed right column $|0\ 0\ 0\ 1\ 0\ 1\ 1\ 1|^T$ is associated with the number $0 \cdot 2^0 + 0 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3 + 0 \cdot 2^4 + 1 \cdot 2^5 + 1 \cdot 2^6 + 1 \cdot 2^7 = 232$. This function is called two and more out of three and can be realized by the Boolean formula

$$c_i' = (c_{i-1} \vee c_i)c_{i+1} \vee c_{i-1}c_i.$$

Function number $90 = 0 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3 + 1 \cdot 2^4 + 0 \cdot 2^5 + 1 \cdot 2^6 + 0 \cdot 2^7$ has the form

$$c_i' = c_{i-1} \oplus c_{i+1}.$$

Note that this function determines a memoryless automaton, because the right-hand side of the corresponding formula is free of variable c_i.

In addition to automaton number 232, let us give another example of an automaton with memory, namely, automaton number $18 = 0 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + 0 \cdot 2^3 + 1 \cdot 2^4 + 0 \cdot 2^5 + 0 \cdot 2^6 + 0 \cdot 2^7$. The next-state function of this automaton can be written as

$$c_i' = !c_i(c_{i-1} \oplus c_{i+1}),$$

where "!" denotes the operation INVERSE.

Consider one more next-state function of an automaton with memory. This function is called one out of three, has the number $22 = 0 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2 +$

$0 \cdot 2^3 + 1 \cdot 2^4 + 0 \cdot 2^5 + 0 \cdot 2^6 + 0 \cdot 2^7$, and is realized by the formula

$$c_i' = !c_i(c_{i-1} \oplus c_{i+1}) \vee !c_{i-1}c_i!c_{i+1}.$$

Function odd parity has the number $150 = 0 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2 + 0 \cdot 2^3 + 1 \cdot 2^4 + 0 \cdot 2^5 + 0 \cdot 2^6 + 1 \cdot 2^7$ and is realized by the formula

$$c_i' = c_{i-1} \oplus c_i \oplus c_{i+1}.$$

Function number $60 = 0 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^3 + 1 \cdot 2^4 + 1 \cdot 2^5 + 0 \cdot 2^6 + 0 \cdot 2^7$ has the form

$$c_i' = c_{i-1} \oplus c_i.$$

Note that this function, as well as function number 90, being independent of one variable, nevertheless, describes an automaton with memory, because, in this case, variable c_i appears in the right-hand side of the formula. Function number $165 = 1 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 0 \cdot 2^3 + 0 \cdot 2^4 + 1 \cdot 2^5 + 0 \cdot 2^6 + 1 \cdot 2^7$ realized by formula

$$c_i' = !(c_{i-1} \oplus c_{i+1})$$

is the inverse of function number 90; this is evidenced by the equality $90 + 165 = 255$.

2. INITIAL CONDITIONS

The structure generated by a cellular automaton with a fixed next-state function is defined by its initial state. In this paper, each automaton at step zero contains a single cell in state 1.

3. INVARIANCE WITH RESPECT TO THE OPERATION EQUALITY

The same behavior of cellular automata with different next-state functions provides a basis for classifying them together. In this case, the EQUALITY invariance operation is applied.

The same class may contain cellular automata having essentially different next-state functions. For instance, trivial behavior (the embryo dies at the very first step) is exhibited by automata with different functions such as identical zero and two and more out of three. The whole class consists of 16 functions: {0, 8, 32, 40, 64, 72, 96, 104, 128, 136, 160, 168, 192, 200, 224, 232}. It is characteristic of these functions that the right columns of their next-state tables have zero in the rows whose left part contains at most a single unit. The transposed columns of their values have the form $|0\ 0\ 0\ ?\ 0\ ?\ ?\ ?|^T$, where the symbol "?" stands for either "0" or "1."

The authors expect that, if two cellular automata of this type exhibit the same behavior in the first five steps, then they continue exhibiting it in the sequel.

3.1. The "Sierpinsky Gasket" Type of Behavior

Automaton number 90 generates a self-reproducing structure called "Sierpinsky gasket." Below, we present the structure generated by this automaton after eight steps:

0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 1 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 1 0 0 0 0 0
0 0 0 0 1 0 1 0 1 0 1 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0 1 0 0 0
0 0 1 0 1 0 0 0 0 0 1 0 1 0 0
0 1 0 0 0 1 0 0 0 1 0 0 0 1 0
1 0 1 0 1 0 1 0 1 0 1 0 1 0 1

Figure 1 demonstrates the same structure after 64 steps.

Automaton number 18 generates exactly the same structure as automaton number 90, since the right columns of their next-state functions differ only in the rows where combinations in the left parts involve two units; however, this cannot occur for the initial conditions mentioned above. Thus, in the case considered, memoryless automata and automata with memory exhibit the same behavior. It should be noted here that the formula that realizes the next-state function number 90 seems to be the simplest mathematical description of the Sierpinsky gasket [5, 6, 8].

Note also that the whole class of Sierpinsky gaskets contains eight automata with numbers {18, 26, 82, 90, 146, 154, 210, 218} = *C*.

It should be observed that this class does not contain automaton number 22. This automaton is unique in its class and generates the modification of the Sierpinsky gasket that consists of zero and unit pedestals. In the structure presented below, adjacent pedestals are differently colored:

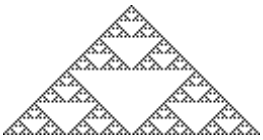


Fig. 1.

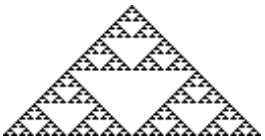
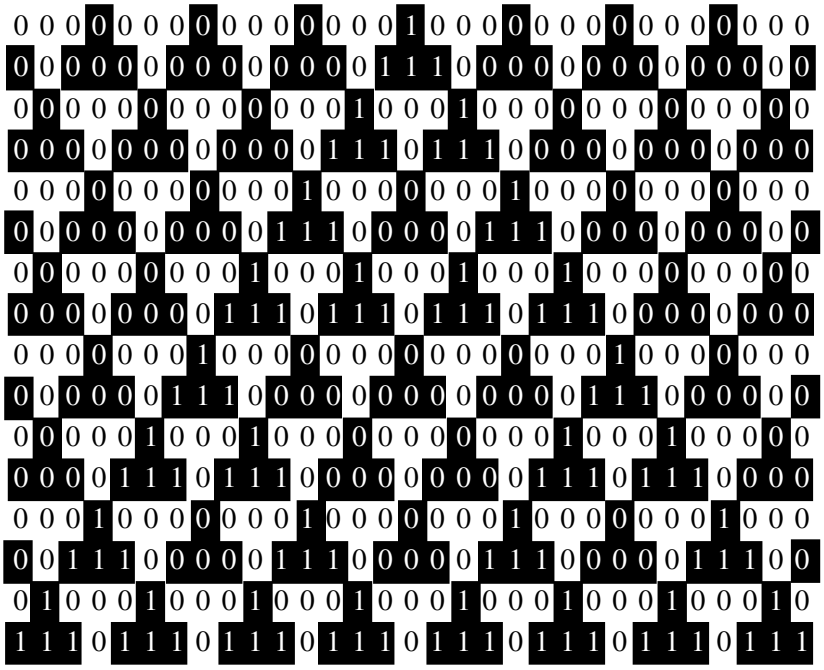


Fig. 2.

Fig. 3.



Replacing zero pedestals with zeros and unit pedestals with units, we obtain the Sierpinsy gasket mentioned above. Figure 2 shows the structure generated by automaton number 22 after 64 steps.

Another unique automaton has number 150. It generates another self-similar structure:

```
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0
0 0 0 0 0 1 0 1 0 1 0 0 0 0 0
0 0 0 0 1 1 0 1 0 1 1 0 0 0 0
0 0 0 1 0 0 0 1 0 0 0 1 0 0 0
0 0 1 1 1 0 1 1 1 0 1 1 1 0 0
0 1 0 1 0 0 0 1 0 0 0 1 0 1 0
1 1 0 1 1 0 1 1 1 0 1 1 0 1 1
```

Figure 3 demonstrates the same structure after 64 steps.

3.2. The "Binary Pascal Triangle" Type of Behavior

There are different ways to construct the Pascal triangle [6, 9–11]. Below, we suggest one more method based on calculating symmetric Boolean functions [12, 13]. It consists of the following steps:

construct a Shannon symmetric multiterminal network [12];

transform it into a symmetric scheme of an algorithm [13];

with the use of Kirchhoff's law, calculate the paths in the scheme constructed; the number of paths to each node of the scheme corresponds to a component of the Pascal triangle.

As an example, Fig. 4 shows a symmetric scheme of an algorithm for five variables, and Fig. 5 shows the structure that reproduces the first five rows of the Pascal triangle.

By the Pascal triangle, construct the binary Pascal triangle

```
1
1 1
1 0 1
1 1 1 1
1 0 0 0 1
1 1 0 0 1 1
```

Rearrange this triangle as follows

```
1
1 1
1 0 1
1 1 1 1
1 0 0 0 1
1 1 0 0 1 1
```

Automaton number 60 generates the rearranged binary Pascal triangle (Fig. 6), which is unique in its class.

3.3. Reproducing Fibonacci Numbers

It is known [10] that left-diagonal cuts of the Pascal triangle reproduce Fibonacci numbers. Let us present an original method for reproducing Fibonacci numbers. This method involves calculation of alternating repetition-free threshold formulas. It is described in [13] and consists of the following steps.

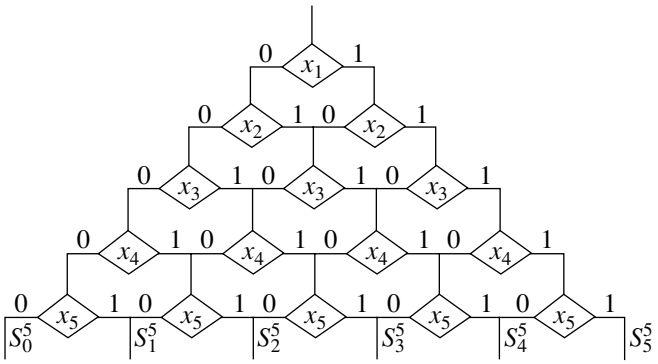


Fig. 4.

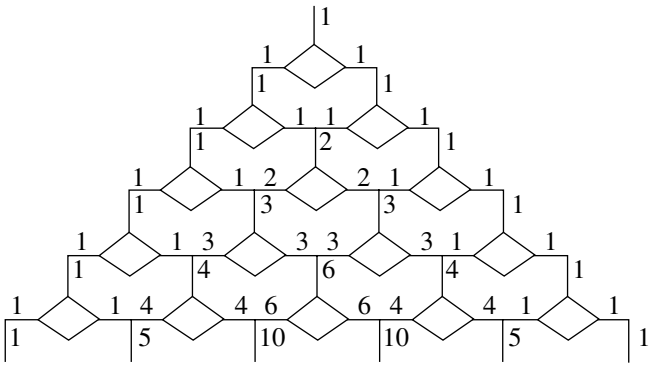


Fig. 5.

By the given formula, construct a linear binary graph;

with the use of Kirchhoff's law, calculate the number of paths from the initial node to the terminal nodes.

Here, the numbers assigned to each conditional node of the graph correspond to the Fibonacci numbers. The number of "unit" and "zero" paths, as well as their total number, are also Fibonacci numbers. Reproduce the Fibonacci numbers by means of calculating the formula

$$f = ((x_1x_2 \vee x_3)x_4 \vee x_5)x_6 \vee x_7.$$

This calculation is illustrated in Figs. 7 and 8.

There are 143 invariance classes with respect to the EQUALITY operation. This number, as well as all the subsequent numbers characterizing different classifications, is calculated with the use of the program mentioned in Section 10. Note that, of these 143 classes, 115 classes consist of a single representative, while the other 28 classes (see Appendix 1) contain two and more representatives. By analogy with classification of the Boolean functions [13], for instance, the PN-classification (abbreviation for Permutation and Negation), we call the above partition into classes E-classification (abbreviation for EQUALITY).

4. INVARIANCE WITH RESPECT TO THE OPERATIONS EQUALITY AND INVERSE

As an example, let us consider the inverse of the structure generated by the automaton number 90. It is generated by the automaton number 165 (Fig. 9). Note that, in this paper, we consider inversion up to the zeroth step.

The above interrelation between "direct" and "inverse" structures can serve as a basis for classifying them together with respect to the invariance operation INVERSE. However, it is unreasonable to classify with respect to the only invariance operation INVERSE, because, in this case, firstly, each class would contain at most two items and, secondly, many items would belong to more than one class. Therefore, the total number of classes could significantly increase.

In view of the above, we suggest that each class assembles structures that are invariant with respect to two operations, namely, operations EQUALITY and INVERSE. With these operations being used, for instance, there is a class that assembles nine automata with the numbers $C \cup \{165\} = \{18, 26, 82, 90, 146, 154, 165, 210, 218\}$.

It is noteworthy that, for the example considered in this section, both the automata behavior and their functions are inverse; this is evidenced by the fact that the sum of their numbers (90 + 165) equals 255. Note also here that the inverses of functions number 18, 26, 82, 146, 154, 210, and 218 do not generate inverse structures for class C. In particular, the behavior generated by function number 37 (the inverse of function number 218) is of little interest (see Fig. 10).

In conclusion to Section 3, note that there are 135 invariance classes with respect to the operations EQUALITY and INVERSE. We call this classification EI-classification (abbreviation for EQUALITY and INVERSE).

5. INVARIANCE WITH RESPECT TO THE OPERATIONS EQUALITY AND MIRROR REFLECT

Let us give an example of the structure that is the twin (mirror reflected with respect to the vertical axis) of the structure generated by automaton number 60. In turn, it is generated by automaton number 102 (see Fig. 11).

There are 89 invariance classes with respect to operations EQUALITY and MIRROR REFLECT. Call this

Fig. 6.

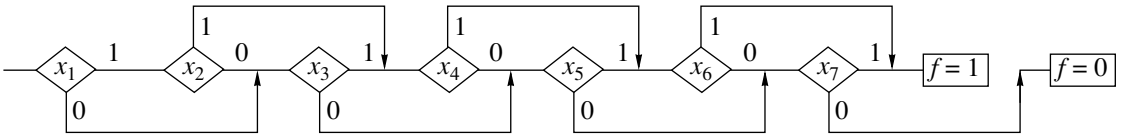


Fig. 7.

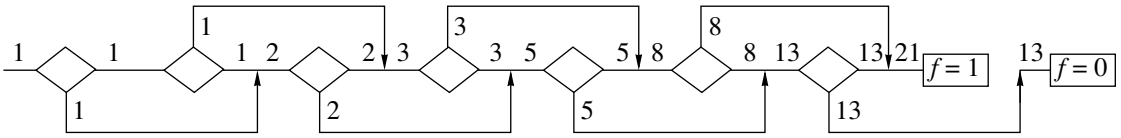


Fig. 8.

Fig. 9.

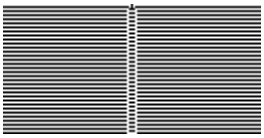


Fig. 10.

classification EM-classification (abbreviation for EQUALITY and MIRROR).

Consider the EM-class with the maximal number of representatives: {2, 10, 16, 24, 34, 42, 48, 56, 66, 74, 80, 88, 98, 106, 112, 120, 130, 138, 144, 152, 162, 170, 176, 184, 194, 202, 208, 216, 226, 234, 240, 248}. This class contains 32 elements, because it unites two E-classes:

{2, 10, 34, 42, 66, 74, 98, 106, 130, 138, 162, 170, 194, 202, 226, 234}, the left diagonal, for which the columns of the next-state table have the form $|0\ 1\ 0\ ?\ 0\ ?\ ?\ ?|^T$;

{16, 24, 48, 56, 80, 88, 112, 120, 144, 152, 176, 184, 208, 216, 240, 248}, the right diagonal, for which the columns of the next-state table have the form $|0\ 0\ 0\ ?\ 1\ ?\ ?\ ?|^T$.

6. INVARIANCE WITH RESPECT TO THE OPERATIONS EQUALITY, INVERSE, AND MIRROR REFLECT

Consider the example of four structures with the numbers {60, 102, 153, 195} belonging to the same class with respect to the operation kit under discussion (see Fig. 12).

It can be seen in this figure that structures number 60 and 102 are twins, as well as structures number 153 and 195. Then, structures 60 and 195 are inverse, as well as structures number 102 and 153.

There are 83 invariance classes with respect to operations EQUALITY, INVERSE, and MIRROR REFLECT. Call this classification EIM-classification (abbreviation for EQUALITY, INVERSE, and MIRROR). It contains the least number of representatives among the classifications considered above.

7. INVARIANCE WITH RESPECT TO THE OPERATIONS EQUALITY AND INVERSE-MIRROR REFLECT.

Consider the same example of structures with the numbers {60, 102, 153, 195}. Under the classification with respect to the operation kit considered, there are two classes, one consisting of two automata with the numbers {60, 153} and the other containing the pair with the numbers {102, 195}. Call this classification E(I + M)-classification (abbreviation for EQUALITY, INVERSE, and MIRROR).

There are 135 invariance classes with respect to the operations EQUALITY and INVERSE-MIRROR REFLECT; this coincides with the number of EI-classes. In fact, these two classifications generate similar classes; only 40 structures are differently classified. They fall into four different classes under EI- and E(I + M)-classifications. The differences between these four classes are demonstrated in Table 1.

All the aforementioned invariance operations can be classified as directly performed (with no offset) operations. Column L0 of Table 2 contains the numbers of classes under the above classifications.

8. CLASSIFICATION WITH ACCOUNT FOR A SINGLE-CELL OFFSET

Introduce new invariance operations, namely, operations OFFSET EQUALITY, OFFSET INVERSE, OFFSET MIRROR REFLECT, and OFFSET INVERSE-MIRROR REFLECT. These operations mean that two structures are examined to mutually correspond either directly or with a vertical, horizontal, or diagonal single-cell offset. Here, structures that corre-

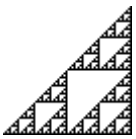


Fig. 11.

spond to each other either directly or with account for any of these single-cell offsets are said to be equivalent.

Classifications with respect to offset operations will be abbreviated similarly to classifications with respect to direct operations, but with the additional letter O (abbreviation for OFFSET).

As an example, consider structures number 20 and 155 (see Fig. 13), which are equivalent with respect to operation OFFSET INVERSE-MIRROR REFLECT. Unlike the above figures, the structures in Fig. 13 are magnified. Structure number 155 is obtained from structure number 20 by means of mirror reflection, inversion, and offset down by a single cell.

There is no point in considering invariance operations with offset by more than a single cell, because the purpose of this paper is to classify together automata with similar behavior, while, in one step, the embryo can affect only the cells that are offset from it by at most one cell.

Offset classifications allow reducing the number of classes as compared to classifications without offset. For instance, EIM-classification distinguishes 83 classes, whereas EIMO-classification reduces their number down to 57 (see Appendix 2). The numbers of classes under classifications with respect to offset invariance operations are presented in column L0 of Table 3.

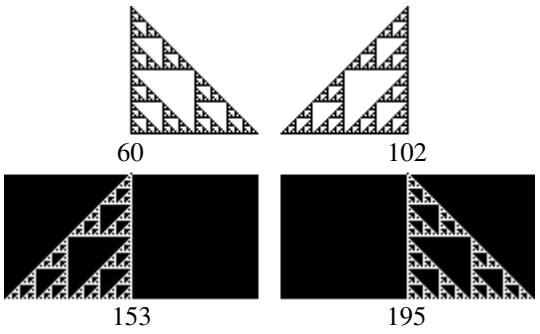


Fig. 12.

Fig. 13.

Fig. 14.

9. CLASSIFICATIONS WITH ACCOUNT FOR LAPSES

It is reasonable that structures are tested for correspondence with respect to invariance operations not exactly, but up to several errors. Indeed, a few cells' lapse is not a reason for classifying the generated structures as unrelated.

Table 1. Difference between EI- and E(I + M)-classifications

EI-classification	E(I + M)-classification
2, 10, 34, 42, 66, 74, 98, 106, 130, 138, 162, 170, 173, 189, 194, 202, 226, 234	2, 10, 34, 42, 66, 74, 98, 106, 130, 138, 162, 170, 194, 202, 226, 229, 231, 234
16, 24, 48, 56, 80, 88, 112, 120, 144, 152, 176, 184, 208, 216, 229, 231, 240, 248	16, 24, 48, 56, 80, 88, 112, 120, 144, 152, 173, 176, 184, 189, 208, 216, 240, 248
60, 195	60, 153
102, 153	102, 195

Table 2. Number of classes in various classifications without offset

	L0	L1	L2	L3	L4	L5	L6
E	143	134	129	128	128	128	127
EM	89	83	81	80	80	80	79
EI	135	124	120	119	119	119	118
E(I + M)	135	124	120	119	119	119	118
EIM	83	76	74	73	73	73	72

Table 3. Number of classes in various offset classifications

	OL0	OL1	OL2	OL3	OL4	OL5	OL6
EO	125	118	118	117	117	117	117
EM	78	75	75	74	74	74	74
EI	88	86	86	85	85	85	85
E(I + M)	88	85	85	84	84	84	84
EIM	56	55	55	54	54	54	54

As an example, consider structures number 233 and 235 (see Fig. 14), which differ only in five cells.

Classifications with respect to operations that account for errors will be abbreviated similarly to classifications with respect to exact operations, but with the addition of Ln (L is the abbreviation for LAPSE and *n* is the maximal number of mismatches).

Columns L0–L6 in Tables 2 and 3 contain the numbers of classes in various classifications *K* with zero to six mismatches. Note that, in the case of no offset (Table 2), all the values in rows EI and E(I + M) coincide, whereas the corresponding values in Table 3 are different.

10. OTHER MATERIALS

The following materials can be found in the Section "Articles" of the Internet sites http://is.ifmo.ru and http://camel.ifmo.ru:

software for simulating and studying automata of the considered type;

software for analyzing and classifying the generated structures;

256 pictures of the generated structures (in Portable Network Graphic format);

256 quadruplicated pictures of the generated structures (in Portable Network Graphic format);

256 text representations of the generated structures;

70 variants of classifications of the generated structures;

generated structures and certain classifications.

CONCLUSIONS

In this paper, various classifications of structures generated by one-dimensional cellular automata from a point embryo are presented for the first time. It is demonstrated that there exist automata with memory and memoryless automata exhibiting the same behavior.

It has already been noted that the investigation of this class of structures has been automated. At the present time, the authors are developing a simulation environment CAME&L (Cellular Automata Modeling Environment & Library), which provides elaborate tools for solving research problems with the help of cellular automata [14, 15]. The environment supports distributed and parallel computations on the cluster platform. For this type of computations, the CTP (Commands Transfer Protocol) networking protocol has been designed [16, 17].[1]

ACKNOWLEDGMENTS

The work was supported by the Russian Foundation for Basic Research, project no. 05-07-90086 and the Borland Corporation.

APPENDIX 1

Equivalence classes with respect to the operation EQUALITY (E-classes) containing more than one element.

(1) 0 8 32 40 64 72 96 104 128 136 160 168 192 200 224 232;
(2) 1 33;
(3) 2 10 34 42 66 74 98 106 130 138 162 170 194 202 226 234;
(4) 3 35;
(5) 4 12 36 44 68 76 100 108 132 140 164 172 196 204 228 236;
(6) 6 38 134 166
(7) 7 19 21 23 31 55 63 87 95 119 127;
(8) 11 43 47;
(9) 14 46 142 174;
(10) 16 24 48 56 80 88 112 120 144 152 176 184 208 216 240 248;
(11) 17 49;
(12) 18 26 82 90 146 154 210 218;
(13) 20 52 148 180;
(14) 28 156;
(15) 50 58 114 122 178 186 242 250;
(16) 70 198;
(17) 81 113 117;
(18) 84 116 212 244;
(19) 129 161;
(20) 139 171;
(21) 151 159 183 191 215 223 233 235 237 239 247 249 251 253 255;
(22) 173 189;
(23) 203 217 219;
(24) 206 238;
(25) 209 241;
(26) 220 252;
(27) 222 254;
(28) 229 231.

[1] You can see and download the CAME&L software on the site http://camel.ifmo.ru.

APPENDIX 2

Equivalence classes with respect to the operations EQUALITY, INVERSE, and MIRROR REFLECT with account for a single-cell offset (EMIO-classes).

(1) 0 8 32 40 64 72 96 104 128 136 151 159 160 168 183 191 192 200 215 223 224 232 235 237 239 247 249 251 253 255;
(2) 1 33 123;
(3) 2 10 16 24 34 42 48 56 66 74 80 88 98 106 112 120 130 138 144 152 162 170 173 175 176 184 187 189 194 202 208 216 226 229 231 234 240 243 245 248;
(4) 3 17 35 49 59 115;
(5) 4 12 36 44 68 76 100 108 132 140 164 172 196 203 204 207 217 219 221 228 236;
(6) 5;
(7) 6 20 38 52 134 148 155 166 180 211;
(8) 7 19 21 23 31 55 63 87 95 119 127;
(9) 9 65 111 125;
(10) 11 43 47 81 113 117;
(11) 13 69 79 93;
(12) 14 46 84 116 139 142 143 171 174 209 212 213 241 244;
(13) 15 85;
(14) 18 26 82 90 146 154 165 167 181 210 218;
(15) 22;
(16) 23 31 55 63 87 95 119 127;
(17) 25 61 67 103;
(18) 27 39 53 83;
(19) 28 70 156 157 198 199;
(20) 29 71;
(21) 30 86 135 149;
(22) 37 91;
(23) 41 97;
(24) 45 101;
(25) 50 58 114 122 178 179 186 242 250;
(26) 51;
(27) 54 147;
(28) 57 99;
(29) 60 102 153 195;
(30) 62 118;
(31) 73;
(32) 75 89;
(33) 77;
(34) 78 92;
(35) 94;
(36) 105;
(37) 107 121;
(38) 109;
(39) 110 124;
(40) 126 129 161;
(41) 131 145;
(42) 133;
(43) 137 193;
(44) 141 197;
(45) 150;
(46) 158 214;
(47) 163 177;
(48) 169 225;
(49) 182;
(50) 185 227;
(51) 188 230;
(52) 190 246;
(53) 201;
(54) 205;
(55) 206 220 238 252;
(56) 222 254;
(57) 233.

REFERENCES

1. G. F. Luger, *Artificial Intelligence: Structures and Strategies for Complex Problem Solving* (Addison Wesley Longman, 1998).
2. J. Von Neumann, *Theory of Self-Reproducing Automata* (University of Illinois Press, London, 1966).
3. T. Toffoli and N. Margolus, *Cellular Automata Machines: A New Environment for Modeling* (MIT Press, Cambridge, Massachusetts, 1987; Mir, Moscow, 1991).
4. S. Wolfram, *A New Kind of Science* (Wolfram Media, 2002), Vol. IL.
5. S. Welstead, *Fractal and Wavelet Image Compression Techniques* (SPIE Press, 2003; Triumf, Moscow, 2003).
6. M. Gazale, *Gnomon: From Pharaohs to Fractals* (Princeton University Press, 1999; Institut Komp'yuternykh Issledovanii, Moscow–Izhevsk, 2002).
7. L. Naumov, "Generalized Coordinates for Cellular Automata Grids," in *Computational Science—ICCS* (Springer, 2003), Part 2.
8. N. Wirth, *Algorithms and Data Structures* (Prentice-Hall, 1986; Nevskii Dialect, St. Petersburg, 2001).
9. M. Gardner, *Mathematical Games, "Scientific American"* (Simon and Schuster, New York, 1964–1969; Mir, Moscow, 1974).
10. R. Bogatyrev, "Golden Triangle," Mir PK, No. 6 (2001).
11. E. Sklyarevskii, "Wonderful Triangle of the Great Frenchman," Hard'n'Soft, No. 10 (2003).
12. C. Shannon, *Selected Works on Information Theory and Cybernetics* (Izd-vo Inostr. Liter., Moscow, 1963) [in Russian].
13. A. A. Shalyto, *Logical Control: Methods of Hardware and Software Algorithms Implementation* (St. Petersburg, Nauka, 2000) [in Russian].
14. L. Naumov, "CAMEL—Cellular Automata Modeling Environment & Library," in *Proceedings of Sixth Internat. Conf. on Cellular Automata for Research and Industry* (Springer, 2004).
15. L. Naumov, "CAME&L—Cellular Automata Modeling Environment & Library," in *Proceedings of XI All-Russian Scientific Conference on Telematika* (CPbGU ITMO, St. Petersburg, 2004), Vol. 1.
16. L. Naumov, "CTP (Commands Transfer Protocol)—Networking Protocol for High-Efficient Computations," in *Proceedings of XI All-Russian Scientific Conference on Telematika* (CPbGU ITMO, St. Petersburg, 2004), Vol. 1.
17. L. Naumov, "Commands Transfer Protocol (CTP)—New Networking Protocol for Parallel or Distributed Computations," http://www.codeproject.com/internet/ctp.asp (2004).