DOI: 10.51790/2712-9942-2020-1-1-4

AUTOMATA-BASED MODEL OF DNA REPLICATION

Georgiy E. Deev, Sergey V. Ermakov
Obninsk Nuclear Energy Institute, National Research Nuclear University MEPhI,
Obninsk, Russian Federation, georgdeo@mail.ru, ermakov@iate.obninsk.ru

Abstract: it is shown that DNA replication can be simulated as computation by the simplest abstract single-state automata: the permutation automata.

Keywords: abstract automaton, B-scheme, DNA (deoxyribonucleic acid), replication, complementarity.

Acknowledgements: this study is supported by RFBR, project No. 20-07-00862.

Cite this article: Deev G. E., Ermakov S. V. Automata-Based Model of DNA Replication. *Russian Journal of Cybernetics*. 2020;1(1):26–30. DOI: 10.51790/2712-9942-2020-1-1-4.

АВТОМАТНАЯ МОДЕЛЬ РЕПЛИКАЦИИ МОЛЕКУЛЫ ДНК

Г. Е. Деев, С. В. Ермаков
Обнинский институт атомной энергетики, Национальный исследовательский ядерный университет «МИФИ», г. Обнинск, Российская Федерация, georgdeo@mail.ru, ermakov@iate.obninsk.ru

Аннотация: показано, что репликация молекулы ДНК может быть промоделирована как вычисление, проводимое простейшими абстрактными автоматами с одним состоянием — перестановочными автоматами.

Ключевые слова: абстрактный автомат, B-схема, ДНК — дезоксирибонуклеиновая кислота, репликация, комплементарность.

Благодарности: работа выполнена за счет гранта РФФИ (проект № 20-07-00862).

Для цитирования: Деев Г. Е., Ермаков С. В. Автоматная модель репликации молекулы ДНК. *Успехи кибернетики*. 2020;1(1):26–30. DOI: 10.51790/2712-9942-2020-1-1-4.

Abstract Automata

The definition of the automaton is comprehensive. An automaton is a compound object that includes the following components:

1. The *set* $Z_k = \{0,1,...,k-1\}$ is the input alphabet of the automaton. It consists of k digits of the base-k number system. The last digit "k-1" should be considered as a single character, although it seems to consist of three separate characters: "k", "-", "1". This is because we do not know how users of the k-base number system may agree on the symbol for the last digit.

2. The *set* $Z_l = \{0,1,...,l-1\}$ is the output alphabet of the automaton. It consists of l digits of the l-base number system. The same observation as for the digit "k-1" is true for the digit "l-1"

The input and output alphabets of the automaton can be different.

3. The *set* $Q = \{q_1,q_2,...,q_n\}$ – is the set of the automaton states. The states can be denoted by any symbols. The states designations (and their hardware implementation) do not affect the automaton operation.

4. The *transition function* $q' = \theta(q,x)$ defines the movement of an automaton from one state q to another q' for a specific input symbol x.

5. The *output function* $y = \sigma(q,x)$ returns the output signal of the automaton in the state q for the input signal x.

Everywhere above $(q',q \in Q,\ x \in Z_k,\ y \in Z_l)$.

For brevity, to specify the components of automaton A, we write:

$$A = A(Z_k, Z_l, Q, \theta(q,x), \sigma(q,x)).$$

Usually, automata with finite sets Z_k, Z_l, Q are considered. Such automata are called *finite automata*.

Just listing the automaton components is not sufficient to define an automaton. The automation operation is to be defined.

The automaton operates in discrete time. We will mark time moments with integers without limiting generality. At an integer point in time, the automaton "comes to life" for a moment, operates and then "freezes" until the next integer point in time comes. In this case, it is governed by the *operating equations* f the following type:

$$A: \begin{cases} q(t+1) = \theta(q(t),x(t)), \\ y(t) \quad = \sigma(q(t),x(t)). \end{cases} \quad (t \in Z = \{\ldots -2, -1, 0, +1, +2, \ldots\}). \tag{1}$$

Other types of the operating equation are also possible.

When an automaton is set to one or another state as the initial state, it will represent a certain mapping.

The effect of the automaton A being in its initial state q, on the object $\bar{x}$ is expressed as $\bar{y} = A\bar{x}|q$.

If all states of an automaton are non-equivalent, then the automaton can represent as many different mappings as many states it has. The range of available representations governs the capabilities of the automation.

Single-State Permutation Automata

We only need single-state permutation automata. A permutation automaton is an automaton with its output function performing a permutation of the output alphabet characters. The number of single-state permutation automata for the quaternary number system Z_4 is 4! = 24.

Single-state permutation automata are defined by type 1 tables.

Table 1

x \ q	0
0	0, y_0
1	0, y_1
2	0, y_2
3	0, y_3

In Table 1 $y_i \in Z_4$ are different. Among such automata, there is an automaton defined by Table 2.

Table 2

x \ q	0
0	0, 0
1	0, 1
2	0, 2
3	0, 3

This automaton implements an identity mapping (identity computation)

$$x_r \ldots x_2 x_1 x_0 \to x_r \ldots x_2 x_1 x_0,$$

i.e. any number $x_r \ldots x_2 x_1 x_0$ translates into itself. In terms of substitutions, it implements an identical substitution

$$\begin{pmatrix} 3\,2\,1\,0 \\ 3\,2\,1\,0 \end{pmatrix}.$$

An arbitrary automaton defined by Table 1, implements an arbitrary substitution

$$\begin{pmatrix} 3 & 2 & 1 & 0 \\ y_3 & y_2 & y_1 & y_0 \end{pmatrix}.$$

It is known how content-rich the algebraic theory of substitution is. As we see, all the facts from the permutation theory can be represented in terms of single-state permutation automata.

Biological Interpretation of the Permutation Automata Operation

In genetics, single-state permutation automata are extensively represented.

The genetics information we used is borrowed from [1].

According to the Watson-Crick model, the DNA molecule as a carrier of hereditary information is two stranded helical nucleotide sequences (double helix) of four types: adenine (A), guanine (G), thymine (T), and cytosine (C). The nucleotides sequence along the molecule is arbitrary [1, p. 22]. The number of nucleotides along the helix is high. The arbitrariness of the nucleotides sequence and their number in the molecule implies a huge number of hereditary information combinations [6, p. 237]. Although the sequence of nucleotides along one helix is arbitrary, the sequence of nucleotides in the parallel helix associated with it cannot be arbitrary and is determined by the distribution of nucleotides in the first helix. This dependent arrangement of nucleotides along the parallel helix is determined by the *complementarity law*: there must be a one-to-one match between nucleotides:

$$A \leftrightarrow T, \qquad G \leftrightarrow C. \tag{2}$$

In the DNA environment, the DNA helices may diverge. Then the lost nucleotides are built on each of them according to the principle of complementarity. As a result, the separated spirals are reconstructed identical to the one that split, and instead of one DNA molecule, we get two identical DNA molecules (not different from the original DNA molecule either.) This process of doubling the DNA molecule is called *replication*, and it (and similar ones that differ in detail, but not in principle) is the foundation of the organism growth.

This is illustrated in Figs. 1 and 2. The captions are taken from [1].

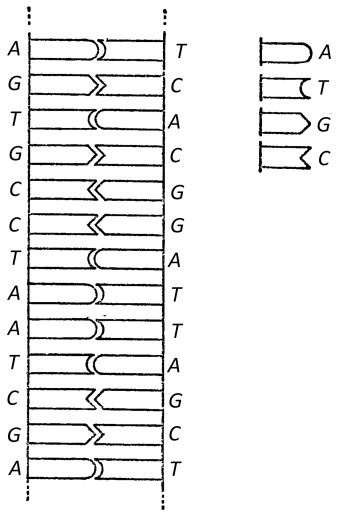

Figure 1. *The DNA molecule looks like a rope ladder consisting of two types of steps: A-T and G-C nucleotide pairs*

The described process of DNA molecule replication can be simulated with the considered single-state permutation automata.

Indeed, both in replication and in other processes that can be interpreted as information retrieval, a certain object moves along the molecule. The object perceives the input information, reacts to it, and produces its output information used to build a molecule.

If we translate this into B-schemes, it can be presented as the diagram in Fig. 3.

The law of complementarity is represented by the links within the rectangle. The dots on the left represent a possible nucleotide sequence. The sequence of dots on the left if read from top to bottom, means that this DNA section has a guanine nucleotide (1) at the top, a thymine nucleotide (2) at the bottom, an

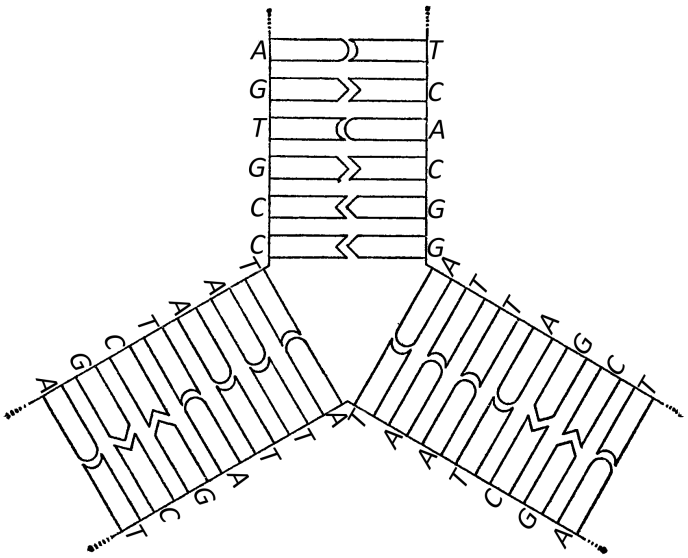

Figure 2. *According to Watson and Crick, the DNA is replicated. As a result, two identical molecules emerge from the original molecule shown in Fig. 1*

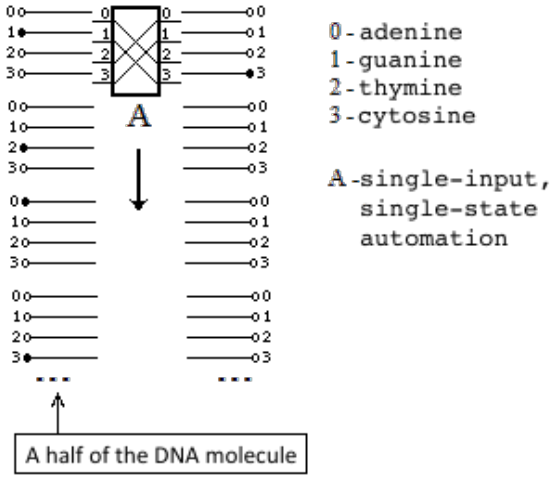

Figure 3. *Automated model of the DNA replication process*

adenine nucleotide (0) further down, and then a cytosine nucleotide (3). On the right almost all points are empty. It means that there is no complementary construction yet.

Automaton *A*, moving along the left half of the DNA molecule, with its left input perceives the input information and uses it to produce an output signal that is complementary to the input. Fig. 3 shows the situation when the automaton already completed the generation of one complementary output, i.e., one complementary nucleotide. Specifically, passing the upper left nucleotide, it perceived the input information through its left input at line 1, "recognized" the guanine nucleotide number 1, and, according to the connections within the automaton (see Fig. 3 and Fig. 4), developed a reaction as the signal 3/, In terms of genetics it means the addition of cytosine nucleotide, complementary to guanine nucleotide, to the DNA molecule. A single-input automaton for complementary overlapping can be represented as an ordinary single-input automaton with a typical *B*-scheme containing four & elements. This is the A_{10} automation (Fig. 4) (The A_{10} automaton is borrowed from the general theory of permutation automata given in [4].)

The field structure of the molecules on the left (Fig. 4), interacting with the field structure of the "rectangle" molecules, creates a new field structure. By the principle of variation, it attracts only one of the four possible nucleotides: the one complementary to the left nucleotide. After attraction, a new field structure is created. It pushes out the molecular formation represented by the rectangle and forces

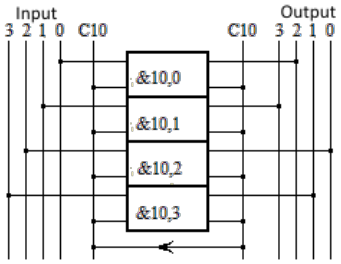

Figure 4. *A_{10} automation for complimentary overlapping*

it to move further along the DNA molecule. After pushing out, the complementary nucleotides are glued together, and the "rectangle" moves on to the next position. There it performs a subsequent interaction with the nucleotide on the left and the process repeats. It should be noted that the field structure of the "left nucleotide-rectangle-right nucleotide" formation is such that the rectangle is always pushed out to one side and grabbed by the next left nucleotide in the molecule. This ensures the directional movement of the rectangle along the molecule. In principle, all these field structures can be estimated. The pushing out effect (always in one direction) can also be found by calculation. It is possible to calculate the process duration and estimate the performance of such a biological computer. As we can see, the simplest, single-state biological automaton does exist. It governs all our life with its incredible complexity. It should be noted that at this elementary abstract level the laws of mathematics allow for various variations such as not 4- but 6-nucleotide formations. However, they are not found on Earth. The question arises: for what reason? Are the principles of variation really to blame?

Be that as it may, biological processes of cybernetic nature can be *directly* simulated by B-computers, as shown above, without any software support, unlike existing computers requiring special software for such simulation.

Conclusion

It is shown that life processes at the DNA molecule level can be modeled by abstract automata represented by B-schemes. The four-letter B-schemes best suit the variational principle of optimization existing in nature.

REFERENCES

1. Frank-Kamenetskii M. *The Live Cell Queen.* M.: AST-PRESS; 2010. 272 p. (In Russ.)
2. Shrodinger Eh. *What Is Life?* AST; 2018. 288 p. (In Russ.)
3. Deev G. E. *Abstract Computers.* M.: Ehnergoatomizdat; 2007. 332 p. (In Russ.)
4. Deev G. E. *The Theory of Computers.* P. 1. Lecture notes. Obninsk; 2009. (In Russ.)
5. De Robertis E. D. P., Nowinski W. W., Saez F. A. *Cell Biology.* 1970.
6. Watson J. *Molecular Biology of the Gene.* M.: Mir; 1978. 720 p. (In Russ.)
7. Kolmogorov A. N. *Automata and Life*. *Matematika – nauka i professiya*. M.: Nauka; 1988. Pp. 43–62. (In Russ.)