Note

Rule Dynamical Property of Genetic System

YOSHINORI NAGAI*,**, MASASHI KITO* and YOJI AIZAWA***

(Received 27 December 2001, revised 31 January 2002)

Synopsis: Rule dynamical property is found in genetic information process by discussing the analogy between information processes in genetic system and computer. We conclude that proteins produced from the nucleotide sequence prints on DNA organize a cell assembly machine and control to each other. The property of rule dynamics is hidden behind protein networks appeared in the cell over time. The proteins presented at any time govern the producing proteins at next time. The changing of protein species just realizes rule change. Hence, rule-dynamical property is revealed.

1. Introduction

DNA has the key information to organize a life system of cell assembly. Following letters of DNA alphabets {A, T, G, C}, we can see the words to produce proteins. We define the genetic system as a system consisting of DNA, RNA, and proteins to organize chemical catalysis networks for the use of substances in the circumstance. A realized system by this manner is called a cell. Proteins regulate organized cells and pull information from DNA. DNA is a library to keep life-plan prints. The interaction between protein and DNA governs the life plan to adapt the circumstance around the life. This just implies rule dynamics. In the present paper, we discuss the feature of gene appearance in the viewpoint of rule dynamics.

Rule dynamics mean the dynamics, of which rule is changed by some physical quantity of system. The rule change is usually seen in the practical systems such as exchange of currencies, neural networks, human relationships so-called complex systems. To see what happen by rule changing, we have constructed a rule-dynamical system for cellular automata (CA) [1]. Cellular automaton systems always have finite number of entire rules as far as every system has finite number of states. Then the temporal change of rules is more easily realized rather than continuous system. If states of a continuous quantity system can be classified into finite number of state groups, rule changing of the system becomes easier controlled. The concept of rule changing is widely acceptable to describe complicated systems. Rule dynamics simply reflect these characteristics of complicated system, while rule dynamics can be constructed by the rigorous method standing on discrete dynamics [2]. General aspect of rules in finite state system is obtained from discrete algebra and discrete functions [2].

* Center for Information Science, and **Faculty of Political Science and Economy, Kokushikan University, 4–28–1 Setagaya, Tokyo 154–8515, Japan, ***Department of Applied Physics, School of Science and Engineering, Waseda University, 3–4–1 Ookubo, Shinjyuku-ku, Tokyo 169–8555, Japan.

We discuss here that genetic system which we mean the producing process of proteins from the information in DNA just behaves like rule dynamics. Since the transcription of nucleotide sequence is regulated by the interaction between proteins and DNA, we regard proteins as control equipments in genetic systems. Usually, a memory segment of computer system is consisting of an address part and a command or data part. Every memory segment of memory equipment can be arranged in free way so that memory equipment of computer can be seen as a one-dimensional array of memory segments. On this viewpoint, a register in control unit pull out a memory segment from the memory unit of a computer. The order of memory segments is discriminated with the address of each segment. In other words, the address of memory segment is identical to the tag attached with transporting material. In this sense, genetic systems also have tags to pull out information from DNA. In Prokaryotes, promoters and operators play like tags [3, 4]. These kinds of regulations are also seen in Eukaryotic cells [3, 4]. Then tags in genetic systems are regulator genes or regulatory sequences. The mRNA in genetic system corresponds to the register state in the computer, but not to register itself, because every register has fixed length while mRNA varies its length. Proteins in a cell form the cell structure and inner-cellular organelles, regulate cell conditions, and control gene information. Thus proteins and mRNAs form the function of control unit in the computer. This corresponded control unit of genetic system is, however, not one-body equipment. The control unit of genetic system is just like multi-parallel processors in computer systems. For these reasons, we can therefore see that the genetic system is the similar system like computer. These points are discussed in section 2.

We apply an analogy of computer for genetic information processes discussed above. If we restrict nucleotide sequences, we can imitate them by discrete cellular automata. The genetic system uses four letters {A, T, G, C} for DNA or {A, U, G, C} for mRNA so that four level cellular automata (CA^4) is appropriate for the modeling. As well known, DNA strands make base pairs such that A–T and G–C by hydrogen bonding, namely, two bonds for A–T pair and three bonds for G–C pair [3, 4]. This bonding feature gives binary property of four letters. In cellular automata (CA) study, temporal change of initial pattern of cell states is observed. In this sense we couldn't find sequence change easily. Hence, the temporal change of base sequences is not direct theme by rule-dynamical CA modeling. The evolution of species caused by base change in DNA is still significant in CA modeling, but it isn't real time activity of lives. When one wants to discuss rule dynamics, the temporal changes of states in a cell are suitable to establish a CA model of rule dynamics. Here "states" in a cell means all the proteins in a cell at time t, by which activities, structures, and functions of a cell are revealed. We consider these points and give brief mathematical descriptions in section 3. The mathematical description for the temporal development of cell state immediately implies rule dynamics brought into genetic system. This aspect leads to the model of CA for the temporal development of organized cell assembly. In this situation, the genetic information processes become inner cell dynamics to bring the temporal change of cell states. We therefore conclude that dynamics realized by proteins is arranged by the temporal appearance of

protein species caused by gene information processes. On the basis of these considerations, we build the model for genetic system with mathematical description in section 3. Finally we briefly discuss central dogma information process in section 4.

2. Correspondence between Genetic and Computer Systems

We had considered the rule-dynamical nature of computer systems [5]. Then it is straight way to compare the processes of appearing genetic information with data processes in computer systems, when we think about the rule dynamics in genetic systems. Genetic information is held in nucleotide sequence of DNA. Then the four base of nucleotides {A, T, G, C} have just the meaning like alphabets. A sequence of bases denotes a sentence of message. In Eukaryotic cells, the sentences to organize an amino acid sequence of protein are embedded into the base sequence of DNA. The meaning of a base sequence comes after splicing processes [4]. The sequence to compose a significant sequence appeared in mRNA are called "exon", and other meaningless sequences locating between exons are called "intron" [4]. The information of mRNA, namely, a sequence of mRNA alphabets {A, U. G, C} is transferred to the sequence of amino acid residues following by codon table [3, 4]. The codon table is just like a discrete function to map four alphabets to twenty kinds of amino acid residues except for start and stop codons. In rigorous sense, the complete description by discrete function requires the mapping from four to twenty two symbols. The process to produce a polypeptide based on the information of mRNA sequence is performed in a ribosome [3, 4]. The triplet code of mRNA is translated by the tRNA (transfer RNA) that bind the amino acid corresponded to a triplet with recognizing triple base of nucleotides [3]. The produced polypeptide forms a 3-D structure with certain fluctuations and then it becomes a protein.

The flow of information to produce a protein from DNA is called "Central Dogma". Francis Crick proposed this concept in 1958. The scheme of central dogma just corresponds to the CPU in a computer. The scheme of correspondence between genetic system and computer system is illustrated in Fig. 1. The both cases of genetic and computer systems show similar flow of information. They have discriminative symbols like letters by which sentences are organized and physical processes are realized. In our aspects, the following matters are corresponded to each other:

1 gene (exon and intron) $\leftrightarrow$ command or data
2 messenger RNA (mRNA) $\leftrightarrow$ sequence of symbols in register
3 protein $\leftrightarrow$ the part of control unit to generate control signal following from symbols in register

The matters described above do not correspond directly in every stage. The produced proteins have different roles. A part of them realize cell structures, inner cell organelles, and function of cell surface. Some of them catalyze chemical reactions with enhancement and inhibition of reaction. One of other parts is the class of proteins to control genes and to make

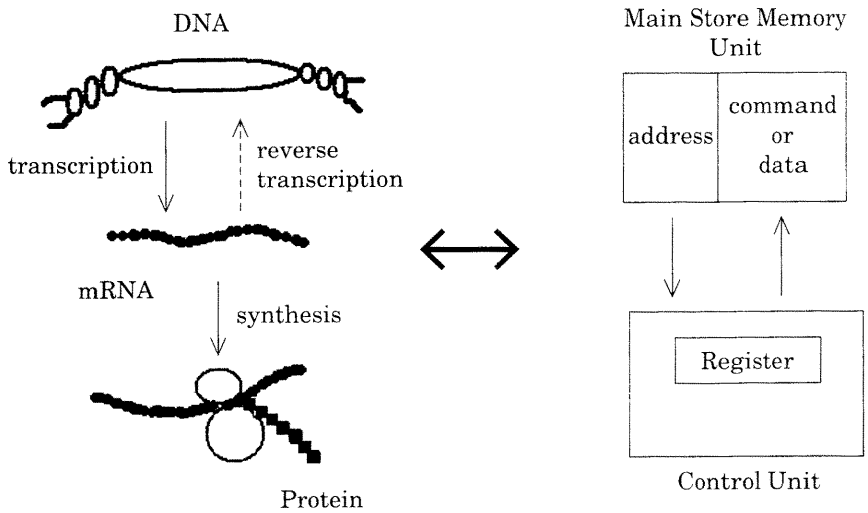

Fig. 1 Analogy between information processes in genetic system and computer.

proteins in ribosome, namely, to control the information processes from DNA to protein. Therefore all the component of living cells are, in principle, constructed by utilizing proteins of which dynamics pull the information in DNA out.

Proteins automatically organize a machine through assembly and catalysis of chemical substances. The interaction between proteins brings the control like machine by producing and dissolving proteins. The role of proteins corresponds to control signals and mechanism of generating control signals like the control unit of computer. The control unit of computer includes register which role is to determine what signal should be generated based on symbols in it. Hence we conclude that the register should be assigned to mRNA. The mRNA denotes which command is activated now from the commands embedded in DNA. In other words, mRNA indicates the active gene in DNA. The assignment of mRNA to symbols in the register should be appropriate. The slight difference of analogy between genetic and computer systems is appeared in correspondence scheme. The right assignment may be that control unit↔(mRNA + proteins).

The main store memory unit of computer can conceptually be divided into address part and command (or data) part [6, 7, 8, 9]. The computer programme (i.e., a sequence of commands) is stored in command part of store memory unit [6, 7, 8, 9]. Following address ordering, each command is processed at the control unit. Taking command flow in the computer into account, memory can be dissolved into slips consisting of the symbols for address and those for command or data. To chain slips, we can get quite well analog of genes, even though the each real gene have no address part in it. The situation is schematically illustrated in Fig. 2.

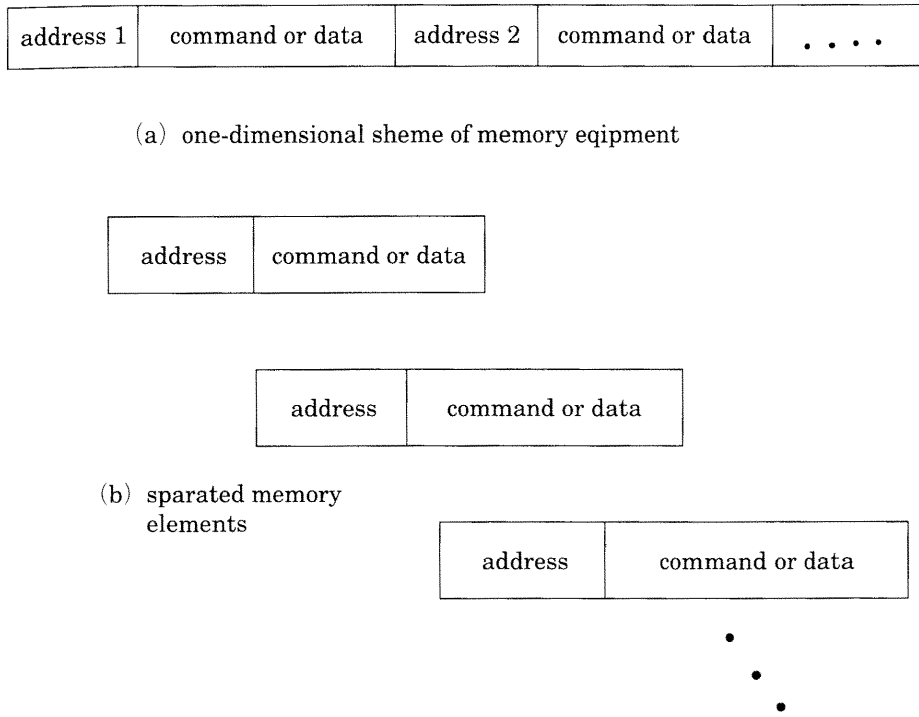

Fig. 2 Slip dividing of memory components.

3. Rule Dynamics in Genetic Systems

We induced an analogy between command flow in the computer information process and the information flow on the protein synthesis from gene in DNA through mRNA. Using this analogy and our previous work to analyze the computer system in the viewpoint of rule dynamics [5], we write down genetic information processes by the following mathematical description:

$$\begin{cases} \mathrm{DNA} = \mathrm{oseq}(1)\oplus\mathrm{gene}(1)\oplus\mathrm{oseq}(2)\oplus\mathrm{gene}(2)\oplus\cdots\oplus\mathrm{oseq}(n)\oplus\mathrm{gene}(n)\oplus\mathrm{oseq}(n+1) \\ \mathrm{mRNA}(k) = \mathrm{Trns}(\mathrm{DNA}, \{\mathrm{protein}(k_1), \mathrm{protein}(k_2), \cdots, \mathrm{protein}(k_m)\}, \{\mathrm{nucleotides}\}) \\ \qquad\quad = \mathrm{Splicing}(\mathrm{gene}^*(k), \{\mathrm{protein}(k_{s1}), \mathrm{protein}(k_{s2}), \cdots, \mathrm{protein}(k_{sz})\} \\ \mathrm{protein}(k) = \mathrm{Ribosome}(\mathrm{mRNA}(k), \{\text{amino acids}\}, \{\mathrm{tRNA}\}), \end{cases} \tag{3.1}$$

where n signifies the total number of genes which are discriminated by assignment of integers, the symbol $\oplus$ means the operation to join two factors A, B, i.e, A B, oseq($\cdots$) denotes the sequence of nucleotides between neighbored genes, gene* (k) is the translated sequence of gene(k) by using alphabets {A, U, G, C}, and Trans($\cdots$), Splicing($\cdots$),

Ribosome(···) mean functions, every which denotes its own function just as mathematical function in abstract meaning. As seen from (3.1), it is open how {amino acids}, {tRNA}, and {nucleotides} are supplied. Then we assume that there exists sufficiently much amount of those substances.

The expression (3.1) is implicit with respect to time. The process of gene information is asynchronous so that time delay appeared in the processing is different for each process. Then it is better to use the population of mRNA and proteins when we describe the process with explicit expression over time. This implies population dynamics of mRNA and proteins, i.e.,

$$\begin{cases} dX(mRNA(k))/dt = G_T(Asite(DNA), X(protein(k_1)), \cdots, \\ \qquad\qquad X(protein(k_m), c(A), c(U), c(G), c(C)) \\ dX(protein(k))/dt = G_R(X(mRNA(k)), X(protein(k_{s1}), \cdots, X(protein(k_{sz}), c(\{amino\ acid\})) \end{cases} \tag{3.2}$$

where c(A), c(U), c(G), c(C) are concentration of nucleotides, X(···)s are population of mRNA and proteins, and c({amino acids}) means the concentration of twenty amino acids, namely, c(Ala), c(Arg), c(Asp), c(Asn), c(Cys), c(Gln), c(Glu), c(Gly), c(His), c(Ile), c(Leu), c(Lys), c(Met), c(Phe), c(Pro), c(Ser), c(Thr), c(Trp), c(Tyr), and c(Val), functions $G_T(\cdot\,\cdot)$, $G_R(\cdots)$ detrmine the rate to produce mRNA and proteins, and Asite(DNA) denote activated site of DNA.

Rule dynamical property is found in the factor Asite(DNA). Rule change implies the change of activated site on DNA. This occurs by the protein behavior of inhibition and enhancement. In other words, Asite(DNA) is the function created with cooperative work of proteins, namely,

$$Asite(DNA) = Asite(DNA, \{proteins\}). \tag{3.3}$$

The expression (3.3) indicates that the rule is changing by proteins, i.e.,

$$gene(k) = Asite(DNA, \{proteins \mid gene(k)\}), \tag{3.4}$$

where {proteins|gene(k)} means the set of proteins to pull the information of k–th gene (gene(k)) out. As known in above expressions (3.3) and (3.4), the index k for gene species is determined by the set of proteins to pick the information of gene(k) up. We call this set of proteins "gene(k) protein set". Then Eq.(3.2) should be written in the following expressions:

$$\begin{cases} dX(mRNA(k))/dt = \delta(k, \sigma(\{proteins\}))G_T(X(protein(k_1)), \cdots, \\ \qquad\qquad X(protein(k_m), c(A), c(U), c(G), c(C)) \\ dX(protein(k))/dt = G_R(X(mRNA(k)), X(protein(k_{s1}), \cdots, X(protein(k_{sz}), c(\{amino\ acid\})) \end{cases} \tag{3.5}$$

where $\delta(k, \sigma(\{proteins\}))$ is the function like Kronecker delta and $\sigma(\{proteins\})$ is the function to give the index number for the gene protein set, i.e.,

$$\delta(k, \sigma(\{proteins\})) = \begin{cases} 1 & \text{for } k = \sigma(\{proteins\}) \\ 0 & \text{for otherwise.} \end{cases} \tag{3.6}$$

The equation (3.5) can formally describe the dynamics of genetic system. Note that the whole activity of a cell requires another set of differential equations to dissolve the mRNA and proteins, to produce tri-phosphate nucleotides and amino acid, and to produce and dissolve other chemical substances which are needed in cell lives. Here we emphasize that rule dynamics is realized on the function $\delta(k, \sigma(\{proteins\})$ to specify the index of activated gene, which select the gene in DNA.

As known from above consideration, we can speculate that the living state of life is protein network with reaction network of chemical substances, while DNA is only holding protein plan prints. DNA and proteins are equivalent matter on the information process because of transfer of symbols in DNA to the symbols in protein. But the resulting functions are quite different. DNA has no machine behavior while proteins just organize a construction like a machine. These are complementary feature. In other words, real feature is protein and virtual feature is DNA.

The induced rule dynamics shown above is different from our previous works on CA. Here we consider what things give CA type rule dynamics with respect to genetic systems. In CA, states (or levels) are assumed to be depending on the system. So we first think about which matter is better to select as the state. In the resent paper, we choose the set protein appeared in a cell at a certain time t as the cell state at the same time. The state of cell is abstractly denoted by S(t). To discriminate each cell, we use suffix j to S, that is $S_j(t)$ for j-th cell (j cell). Every cell has a set of proteins so that it is needed to represent protein set explicitly. Then the j cell state $S_j(t)$ means that $S_j(t) = \lambda(\{proteins\}_j(t))$, i.e., the integer assigned the set of proteins in j cell at the time t, where $\lambda(\cdots)$ is the assignment function of protein set to integer. Hence we know that a choice is $S \in \{0, 1, 2, 3, \cdots, \max\#(\{proteins\})\}$, where $\max\#(\{\cdots\})$ denotes the number of elements for maximum protein set appeared in the cell. This choice of state expression leads to the direct model of CA for the genetic system. Let maximum integer given by $\max\#(\{proteins\})$ be L. Then we can treat genetic system as the L level cellar automata (CA^L). The number of neighbors around a cell, which affect to the target cell, is open for the modeling. If we choose three neighbors, a model can be organized on L-level 3-neighbor CA (CA_3^L). Then the rule dynamics on CA_3^L is written as follows:

$$S_j(t+1) = \Sigma_r \delta(r, \sigma(\{S_j(t)\}) F_r(S_{j-1}(t), S_j(t), S_{j+1}(t)), \tag{3.7}$$

where $\delta(r, \sigma(\{S_j(t)\})$ is the Kronecker delta like function defined in eq.(3.6), and F_r is the r-th discrete function of L-levels and 3-neighbors (the manner to construct discrete function see ref.[5]). In the equation (3.7), the rule number of $F_r(\cdots)$ is changed by state of cells $\{S_j(t)\}$ temporally, which means rule dynamics.

The rule dynamics described by eq.(3.7) is a model to organize a individual life or a differentiation model of cell assembly. Since the state indicate a family of protein set, the cell

state should be related to the rule to realize the differentiation of cells after cell division. In the present model given by eq.(3.7), the states have a path way restricted by the differentiation of embryo, namely, a protein set y definitely comes after the protein set x. This means that the deterministic causality $S(t+1)=f(S(t))$ exits there. The j cell is influenced neighbors j -1 and $j+1$ for three-neighbor model. This effect means that $S_j(t+1)=f(S_j(t))\Rightarrow S_j(t+1)=f^*(S_j(t))$, namely, the protein set causality is modified by the interaction between the target cell and its neighbors. The equation (3.7) implies these matters. Another expression to describe these matters is that

$$S_j(t+1)=\Sigma_r\delta(r,\ \sigma(S_{j-1}(t),\ S_j(t),\ S_{j+1}(t)))f_r(S_j(t)). \tag{3.8}$$

The model described by eq.(3.7) or (3.8) performs the process in the synchronous way. But the actual process in the living cell occurs at asynchronously. Then a certain time interval Δt should be introduced to describe the asynchronous processes. This time interval is determined by the states $S_{j-1}(t)$, $S_j(t)$, and $S_{j+1}(t)$, i.e., $\Delta t=\phi(\{S_{j-1}(t),\ S_j(t),\ S_{j+1}(t)\},\ S_j(t+\Delta t))$. Then the asynchronous model becomes the implicit form of difference equation, i.e.,

$$\begin{cases} S_j(t+\Delta t)=\Sigma_r\delta(r,\ \sigma(\{S_j(t)\})F_r(S_{j-1}(t),\ S_j(t),\ S_{j+1}(t)), \\ \Delta t=\phi(\{S_{j-1}(t),\ S_j(t),\ S_{j+1}(t)\},\ S_j(t+\Delta t)). \end{cases} \tag{3.9}$$

The asynchronous nature introduced here is not essential when one considers the rule-dynamical property of genetic system. The consideration of asynchronous nature is to adapt the reality of the model. The essential point is that the rule dynamics is the change of processes in a cell which is caused by the influence from the surrounding cells

4. Discussion

Following the analogy between computer and genetic systems, we can call the information flow of genetic and computer systems "central-dogmatic type of information processes." This type of information process has a print in order to induce the dynamics by central information. The flow of information is one way from the central information to controlled dynamics. Although the back signal from control equipment to store memory unit is exist in the CPU, the information brought by the programme (or commands) seems to be one way from memory to control only. Usually the back signal is the data flow to stored memory. If the command goes back to other place of store memory, new information process is induced by it. But this does not mean that store memory unit suddenly turns into control unit. This aspect for the information flow leads to an assertion that the information is the realization of dynamics from the symbols. In other words, the information is controlled dynamics appeared in a system. The control is the key point whether the dynamics is information process or not. Hence we can image another type of information process caused by dynamics. In the present aspect, control means the causality of between one dynamics and another dynamics. If the dynamics X always bring dynamics Y after X, the chain of dynamics X and Y just

means control. If the definite chain of dynamics A, B, C, D, ··· is realized by some interactions in a system, the chain of dynamics undergoes information process. If the chain can describe some static states, the array of static states realizes the programme. DNA is a kind of static states. The quasi-stable states can also realize a programme nature. The consideration here leads to a speculation that another type of information process without programme can exist in the natural world. No programme information process organizes self-induced manner of dynamics. This type of information process also lies in the category of rule dynamics. These ideas are also applicable for brain function (or information process in the brain). It is very interesting whether or not brain information processes have programmes. In other words, the brain information process is central dogma type or not.

References

[1] Y. Nagai and Y. Aizawa, "Rule-dynamical generalization of McCulloch-Pitts neuron networks", BioSystems, **58** (2000) 177–185.
[2] Y. Nagai, T. Maddess, and A. Ankiewicz, "Discrete algebra on cellular automata and binary textures", Mem. Kokushikan U Cent. Inform. Sci. **22** (2001) 51–64.
[3] B. Alberts, D. Bary, J. Lewis, M. Raff, K. Roberts, and J. D. Watson, Molecular Biology of The Cell, (1983) Garland Publishing, New York & London.
[4] L. H. Hartwell, L. Hood, M. L. Goldberg, A. E. Reynolds, L. M. Silver, and R. C. Veres, Genetics, (2000) McGraw Hill, USA.
[5] Y. Nagai, H. Yamaguchi, A. Ichimura, and Y. Aizawa, "Ruledynamical Nature of Computer Processes and Communications", Mem. Kokushikan U Cent. Inform. Sci. **21** (2000) 62–77.
[6] J. P. Hayes, Computer Architecture and Organization, 2nd ed, McGraw-Hill New York, 1988.
[7] D. Cassel and M. Jackson, Introduction to Computers and Information Processing, 2nd ed, Reston Publishing, Reston, Verginia, 1985.
[8] J. Ferguson, Microprocessor Systems Engineering, Addison-Wesley Publishing, Wokingham, England, 1985.
[9] K. Tezuka, Foundation of Electric Computer (in Japanese), Shokohsha, Tokyo, 1978.