Memoirs of the Kokushikan Univ. Center for Information Science. No. 21 (2000)

Paper

Ruledynamical Nature of Computer Processes and Communications

YOSHINORI NAGAI[1,2], HIDEKI YAMAGUCHI[3], ATSUSHI ICHIMURA[1], and YOJI AIZAWA[4]

(Received 14 Jan. 2000, revised 31 Jan. 2000)

Abstract: The concept of ruledynamics on the two state cellular automata of neighborhood-three (CA_3^2) is used to consider the nature of computer systems and their communications. Both computer system and ruledynamics on CA_3^2 are binary systems of information processing. In the present paper computer communications are reduced to the interaction of two ruledynamical systems where partial bit arrays are exchanged. In the interacting ruledynamical systems, only one bit swapping of memory state gives rise to the difference on the temporal change of one-dimensional bit array from non-interacting ruledynamical systems. Every ruledynamical system has strong tendency to hold its proper characteristics. This tendency apperas even in the case of bits almost swapping memory states. If one uses this kind of computer processor, every computer system connected to an internetwork of computers continue to maintain its intrinsic capabilities.

1. Introduction

We have been studying ruledynamics on two state cellular automata of neighborhood-three(CA_3^2) from several viewpoints[1-6]. Since the ruledynamics on CA_3^2 are organized in binary systems, it is obvious that the ruledynamics maintain their relation to computer systems. We considered how a ruledynamical system carries out the ordinal computaion performed by usual computer systems[4]. In the present paper, we shift the viewpoint from how a ruledynamical system performs standard computer capabilities to what class of ruledynamics a computer system can perform. To see the temporal change of whole binary elements in a computer, we know a von Neumann type computer[7] realizes a kind of ruledynamics. A temporal change of whole binary elements is equal to a pattern dynamics (temporal change of cell states) in a cellular automata system. The pattern dynamics lies on the stage of cellular automata. Following the works by Wolfram[8], a temporal pattern of binary elements is generated by a rule which uniformly applies over time development. Here the rule means the determination procedure for the state of a site at a time from the cell states of two nearest neighbors and self site at a previous time. The number of whole rules in CA_3^2 is 256[8].

---

[1] Center for Information Science

[2] School of Polotical Science and Economic Sciences, Kokushikan University, 4-28-1 Setagaya, Setagaya-ku, Tokyo 154-8515, Japan,

[3] Department of International Bussiness and Information, College of International Relations, Nihon University, Bunkyo-cho 2-31-145, Mishima-shi, Shizuoka-ken 411-8555, Japan,

[4] Department of Applied Physics, School of Science and Engineering, Waseda University, 3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555, Japan

Aizawa proposed the concept of ruledynamics and we realized a system of ruledynamics on CA_3^2[1]. We have developed the concept of ruledynamics and through this have been recognizing its broader nature. On the basis of our continuing consideration for ruledynamics, we discuss computer systems and computer communications, here. The ruledynamical nature of a computer is determined by its processor. The control part in the Central Processing Unit (CPU)[9] gives rise to the change of binary state of register elements. The control part is essentially a ruledynamical system within the computer. The memory part of the CPU has few ruledynamical properties because it acts passively. Details are discussed in the next section.

In section 3, we discuss computer communications from the ruledynamical point of view. In a general sense, computer communication means data transportation from one computer to others. We simplify the definition of computer communication to the interchange of the partial binary data stored in each computer. Since the state changes occur on the control part of the CPU mentioned above, computer communication implies the interaction between the registers in computer systems. If the interaction between registers brings about a change of every register state, the interaction causes a kind of calculation. This is the same kind of computation as in a multi-processor system or a convection machine. Large scale computer communications appear in the internetworks of computers. The server role computer on each site of a internetwork acquires data from other server machines in the internetwork by giving signals of a data sequence into the communication lines in the expected manner (TC/IP protocol et. al.)[10]. In this sense, the relationship between computers is similar to that between control and memory parts of a CPU. On the other hand, the computers are changing their states by interaction with network lines. On the focus position of interaction between computers and internetwork lines, computers is temporally being changed their states by the partial interchange of binary data between them.

This situation is similar to the interaction of registers. The computer communication can be reduced to the interacting ruledynamical systems, where a part of binary data of both ruledynamical systems is exchanged in each step of the computer processes. A class of rules determines non-independent processes of registers. Some kind of rule group gives independent processes. These depend on the construction of an interacting ruledynamical system. How the ability of the interacting ruledynamical system depends on the architecture of the system is a problem for consideration in the future. There is no easy solution for this question.

2. Computer System and Ruledynamics on CA_3^2

Even if the computer system has developed and differs from the original type of electric computer proposed by von Neumann (von Neumann type computer)[11], the computer systems still falls within the category of automata systems. They use any physical element as a two-state one, namely, a binary element. The conceptual scheme of von Neumann type computer is presented in Fig. 1 modified from Ref.12. The computation is brought about by the interaction between the main memory and control units, because any arithmetic calculation can be performed by using a computer program if the arithmetic logic unit is removed. Input and output equipments are set for humans to use the computer system. When one dicusses any process occuring in a computer, it is sufficient to treat the sub-system consisting of main memory and control units as a single computer system. This sub-system is equivalent to a Turing machine[13] illustrated in Fig. 2. In a Turing machine, the tape stores programs and data, and the machine initiates a process by reading from and writing to the tape, together with undertaking a change of binary array on the registers.

Our perspective is around the Turing machine, but we will unite the main memory and control parts into a finite number of binary elements, where each binary element takes two states $\{0, 1\}$. Before getting the ruledynamical property of these united systems, we see that a model scheme of computer systems gives the nature of ruledynamical systems. Figure 3 shows the temporal change of states for binary elements of a memory and a register, i.e., the pattern dynamics of binary cells which are assigned to a memory part and a register in the control unit of the CPU. The wide belt of white and black squares in the left side of Fig. 3 denotes the pattern dynamics of the memory part, and two narrow belts of white and black squares in the right side of Fig. 3 signify the states of the register at the time immediately after reading the binary digits from the tape (left narrow belt) and those at the time after change of the register state obeying inner rules of the register (right narrow belt). As seen from Fig. 3, the memory part is identical to the type II behavior of CA_3^2 which is mentioned by Wolfram[8]. A part of the sites in memory cells, which are treated at the same time by the register, also change the states depending on their binary digits array. Other parts in the memory cells hold their binary digits array. This can be realized by using the identical transformation of states. Since Wolfram rules[8] have the identical transformation[5], the ruledynamics can carry out to change the state of whole elements at every time step to imitate the temporal change of binary elements in the computer system. Hence, we conclude that the state change of the process in a computer means the partial rule change occured on CA_3^2.

In the ruledynamical point of view, a part of sites on CA_3^2 performs the state change using the properly defined rule at a time, and remaining part of sites on CA_3^2 uses the rule for identical transformation. We therefore classify the computer system into a sub-group of site-dependent rule-change ruledynamics on CA_3^2[6].

We usually investigate ruledynamics on CA_3^2 using intermittent like temporal rule changes[1-6]. Then the intermittent like temporal rule changes is introduced to the state change at the memory in a register (register memory). The computer simulation for the CA_3^2 model computer is shown in Fig. 4. The register memory size in the simulation is 5. This computer simulation cannot realize the intermittent nature of ruledynamics. The reason is the smallness of the register memory size. Since the ruledynamics of intermittent like temporal rule changes is carried out only on the register memory, we investigated the dependence on the length size of register memory. The results are shown in Fig. 5. As seen from Fig. 5, the

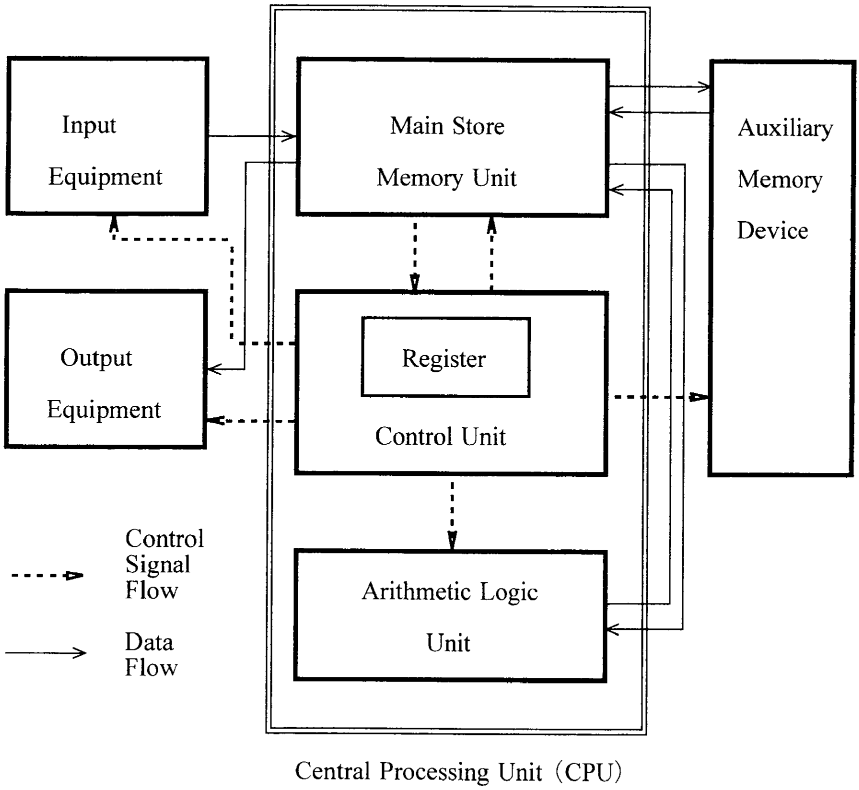


Fig. 1 Schematic Illustration of von Neumann Type Computer

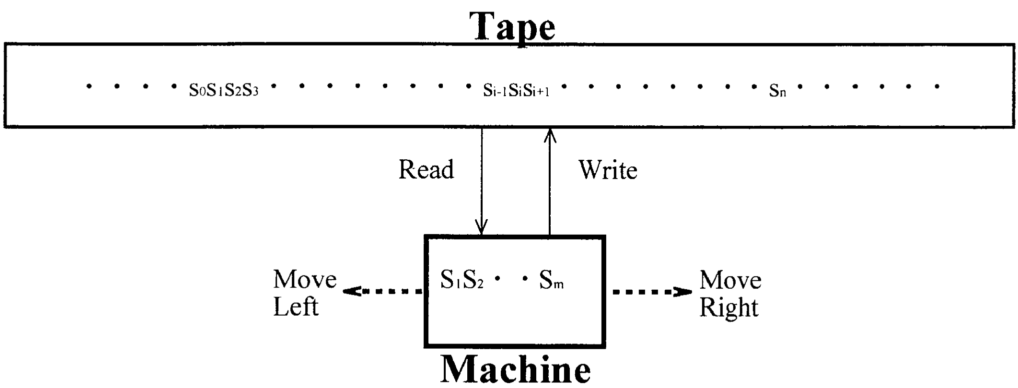


Fig. 2 Scheme of Turing Machine

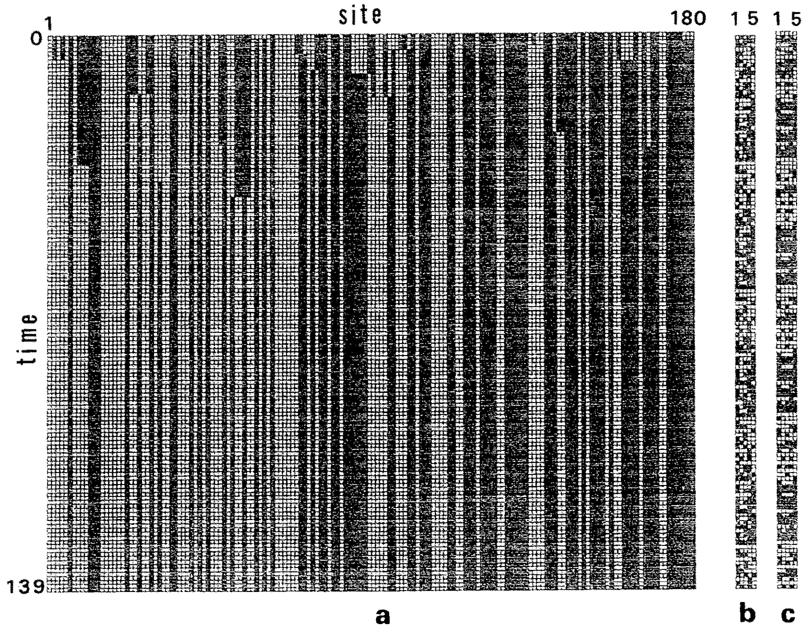


Fig. 3 Temporal Pattern of Memory and Register States in A Simplified Computer Model
a: Main memory state changes, b: Register state changes at the times immediately after reading data, c: Register state changes at the time after state transitions by inner intrinsic rules. The register length size is 5 memory elements and the main memory size is 180 memory elements.

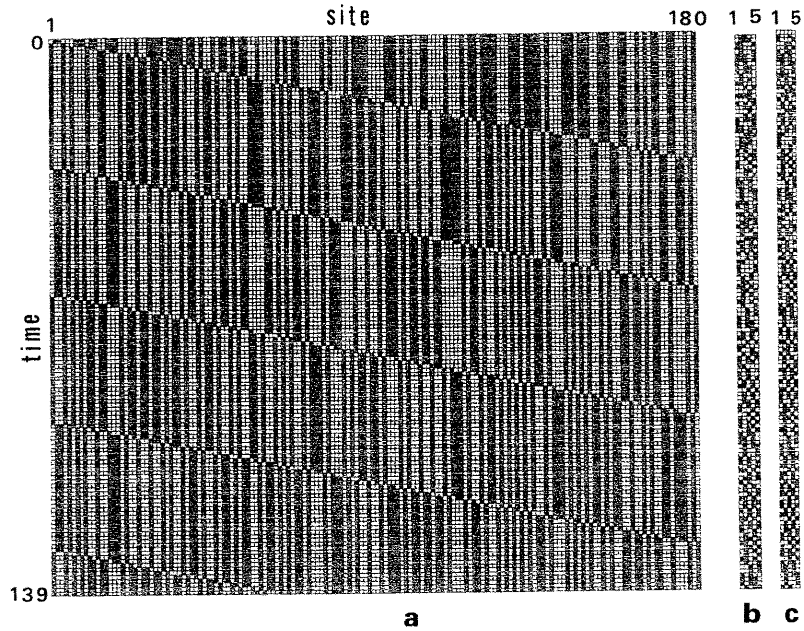


Fig. 4 Temporal Pattern of Memory and Register States for Intermittent like Ruledynamics
The a, b, c are the same as stated in Fig. 3. The register and memory length sizes are also the same as stated in Fig. 3. Rule change parameters are the same ones as stated in Fig. 6.

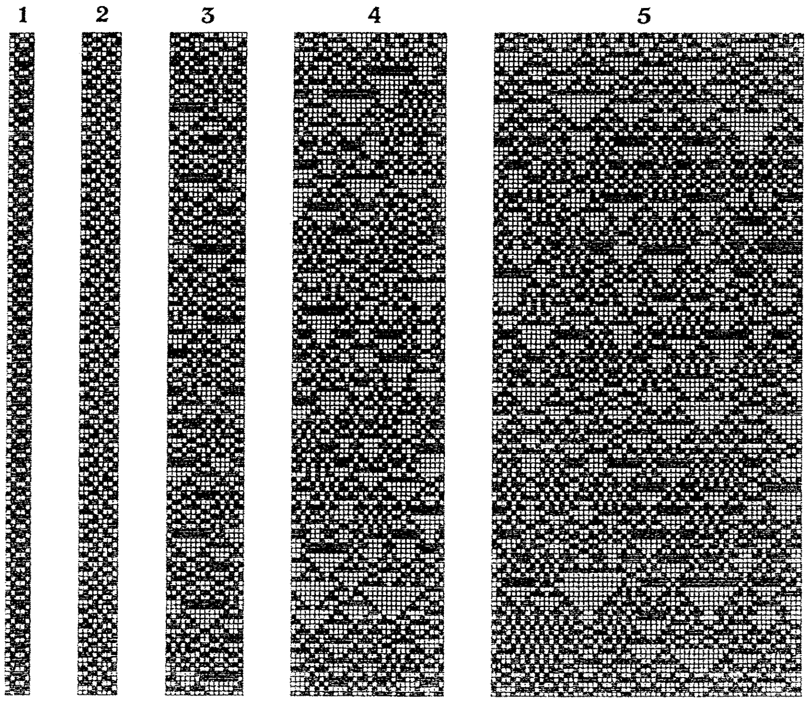


Fig. 5 Register Length Dependence of Ruledynamical State Change
The register length size are 1 : 5, 2 : 8, 3 : 16, 4 : 32, 5 : 64. Rule change parameters are the same ones as stated in Fig. 6.

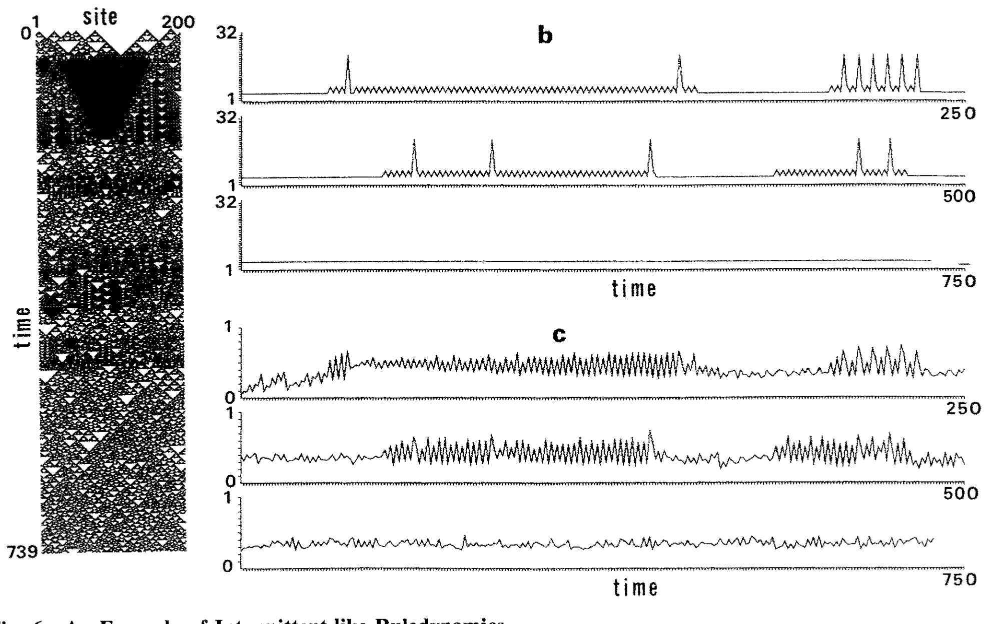


Fig. 6 An Example of Intermittent like Ruledynamics
a: Temporal pattern of cell states, b: Rule changes over time, c: Time course of average activity. The parameter values are $\eta_1=-1$, $T_1=0.65$, $\eta_2=-1$, $T_2=0$, $\eta_3=-1$, $T_3=1$, $\eta_4=1$, $T_4=0.45$, $\eta_5=-1$, and $T_5=0$.

lengthening of memory size recovers the intermittent like nature of a ruledynamics as shown in Fig. 6. Note that the intermittent like rule used here is expressed as following equations[2,3], namely,

$$S_i(t+1)=\Sigma\theta(\eta_k(\langle S_i\rangle_t-T_k)g_k(S_{i-1}(t),\ S_i(t),\ S_{i+1}(t)),\ \mathrm{mod}\ 2, \tag{2.1}$$

where $\langle S_i\rangle_t \in [0,\ 1]$ means the average activity of ruledynamical system, $T_k \in [0,\ 1]$ is a threshold value for k-th fundamental rule switching, and $\eta_k=\pm 1$ is the coefficient for the switch on for lower value of the threshold or that for upper value of the threshold according to $+1$ or -1, respectively. The number of fudamental rules are five[1-6].

We have seen that the computer is a ruledynamical system. Then the modeled system can be described with reccurence type difference equations. We introduce the state vector $S(t)$, which denotes the binary digit array of the memory part at time t, and the register state vector $S_r(t)$ at time t. Futhermore we divide the state vector of memory part $S(t)$ into the summation of register size of memory vector $S_i^m(t)$ ($i=1,2,\cdots$,n), namely,

$$S(t)=S_1^m(t)\oplus S_2^m(t)\oplus\cdots\oplus S_i^m(t)\cdots\oplus S_n^m(t). \tag{2.2}$$

Using the decomposition of memory part, the model of a computer system is written as follows:

$$\begin{aligned} &S_r(t)=G_r(S(t))(\text{means } S_r(t)=S_k^m(t),\ k=\Psi(S_r(t))),\\ &S_r(t+1)=\Sigma\varepsilon_\chi(S_r(t))F_\chi(S_r(t)),\\ &S(t+1)=G_w(S(t),\ S_r(t+1))(\text{means } S_j^m(t+1)=S_r(t+1),\ j=\Phi(S_r(t+1))), \end{aligned} \tag{2.3}$$

where F_χ denotes the state transition of register with χ-th rule, ε_χ means that χ-th rule is switched on or off depending on the register state vector, and G_r and G_w signify read and write, respectively. $\Psi(S_r(t))$ and $\Phi(S_r(t+1))$ are the functions to determine a reading region of the main memory and a writing region of that, respectively. This description gives a closed form of expression for memory state vector $S(t)$, i.e.,

$$S(t+1)=G_w(S(t),\ \Sigma\varepsilon_\chi(G_r(S(t)))F_\chi(G_r(S(t)))). \tag{2.4}$$

This expression implies that multi-mapping yields the ability of general Turing machine. Usually, functions are not analytical[14]. If we can give analytical forms of these functions, any problem can be solved analytically. The ruledynamical nature is included in the expression $\Sigma\varepsilon_\chi(S_r(t))F_\chi(S_r(t))$. Then we can explicitly assert that the computer system is a ruledynamical system.

3. Computer Communication and Interaction Between Ruledynamical Systems

As stated in the introduction, we reduce the computer communication to the partial interchange of binary states in the interacting register memories. In the actual computer net-

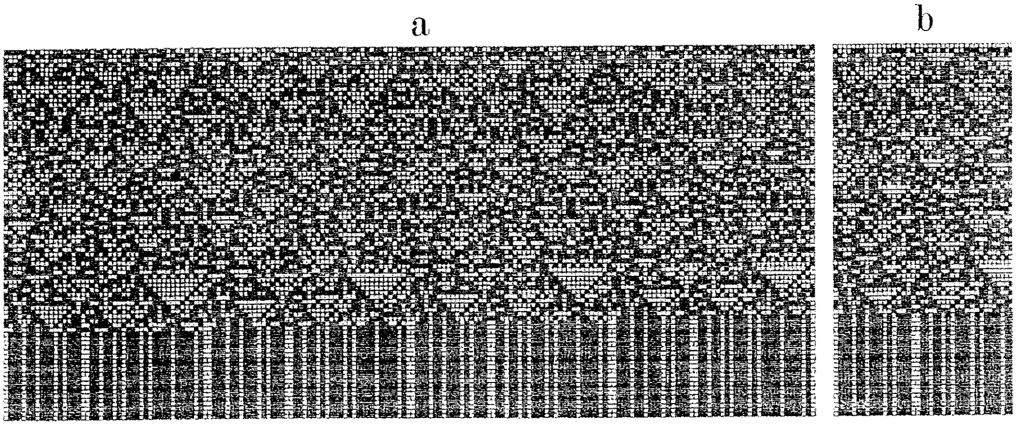
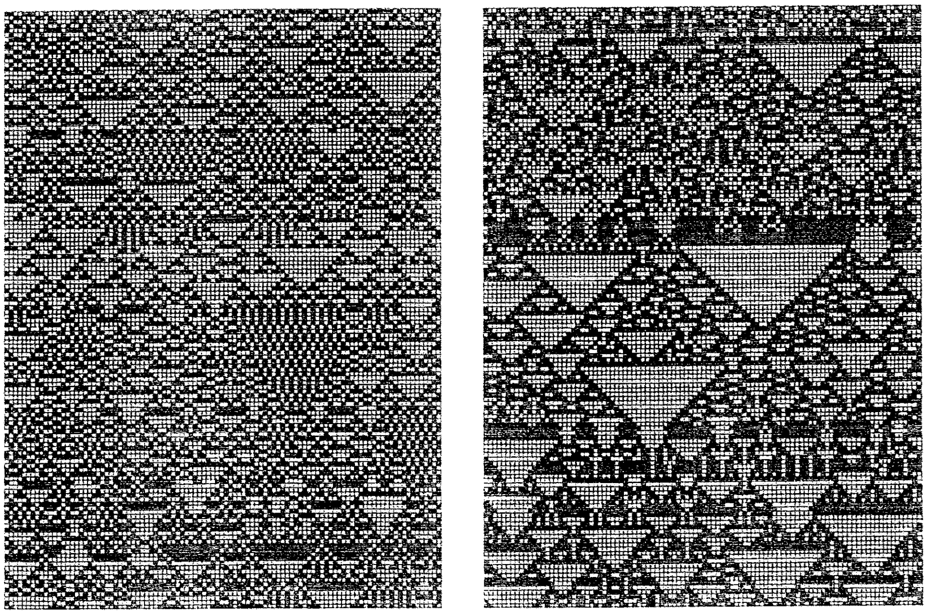


Fig. 7 Stroboscopic Sampling Temporal Pattern of 32 Register Length for Intermittent like Rule Changes
a: Main memory, b: Register states at the times after transitions obeying intermitent like ruledynamics. The rule change parameters are $\eta_1=-1$, $T_1=0.65$, $\eta_2=-1$, $T_2=1$, $\eta_3=-1$, $T_3=0$, $\eta_4=1$, $T_4=0.45$, $\eta_5=-1$, and $T_5=0$. The register length is 32 and the size of memory is 180.

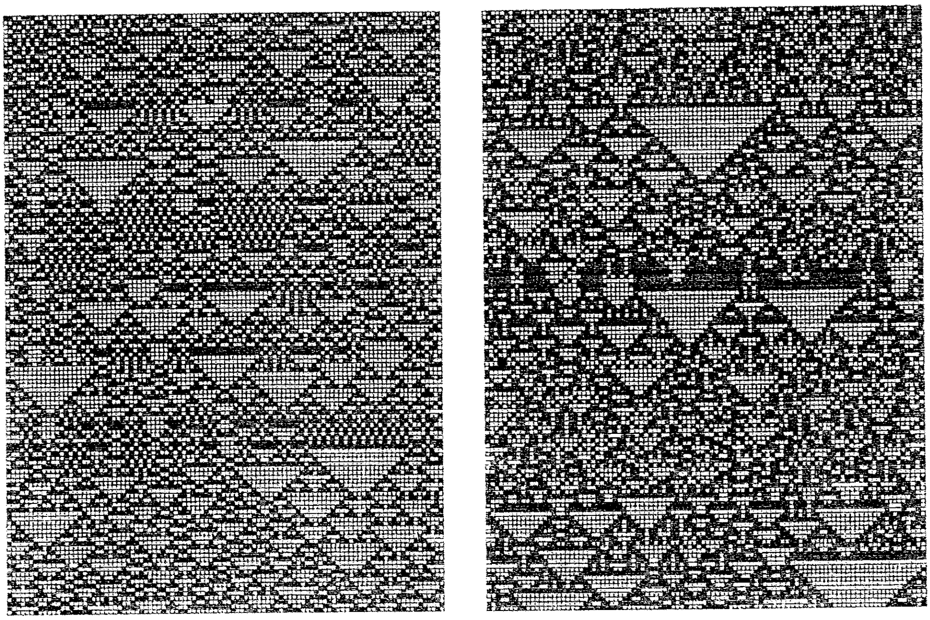
work, the sending and receiving data are temporarily stored on some kind of bus memory for communications, and then they go to the main memory or other memory followed by their characteristics[9]. Concerning with ruledynamical viewpoint, the temporal change of the rule occurs at the register memory in the control unit. The data flow of the round trip between main memory and register memory yields the ruledynamics in the main memory. The stroboscopic sampling of main memory states over time coincides with the temporal change of pattern in a ruledynamics. An example is shown in Fig. 7. Hence, the reduction of computer communications to the interaction of ruledynamical systems is reasonable.

Here, we treat an interacting ruledynamical system of two register memories. The interaction in the system is the partly swapping of register memory states. We discuss the two register communications to show what kind of variation in pattern dynamics appears by changing the size of swapping memory area. The examples are shown in Figs 8 and 9. We use two different ruledynamical systems. In both systems, the rule change is caused by the change of the mean state value averaged over the memory element states in the register. Each memory element takes a state value 0 or 1 at a time. The procedure of rule changes is followed in previous of our works[1-6]. Every ruledynamical system performs the change of states according to its intrinsic properties of ruledynamics. The pattern dynamics for the state of memory elements is therefore different to each other. By the memory swapping interaction, the pattern dynamics become different from those in non-interacting case. Kim and Aizawa[15] showed that the ruledynamics, in which the rule is changed by the average of state values, have the synchronization feature for the average activity of the system. This implies that the actual pattern dynamics is different for different initial binary digit arrays. The average synchronization is meant in a statistical sense.

As seen from Fig. 8, one bit data swapping gives rise to the different pattern dynamics from the pattern dynamics in non-swapping case. For Figs 8 and 9, we used the same initial

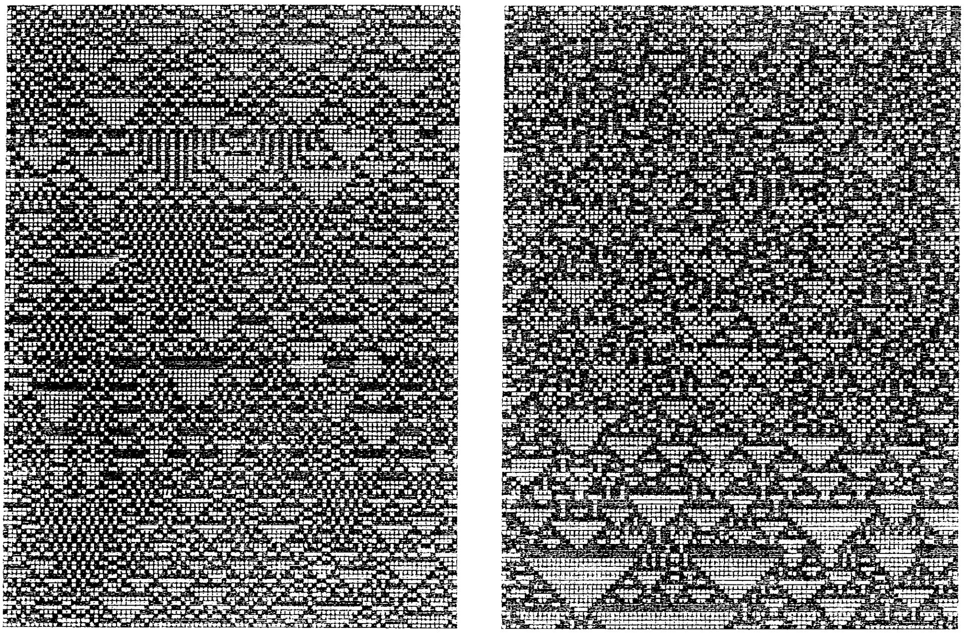


A

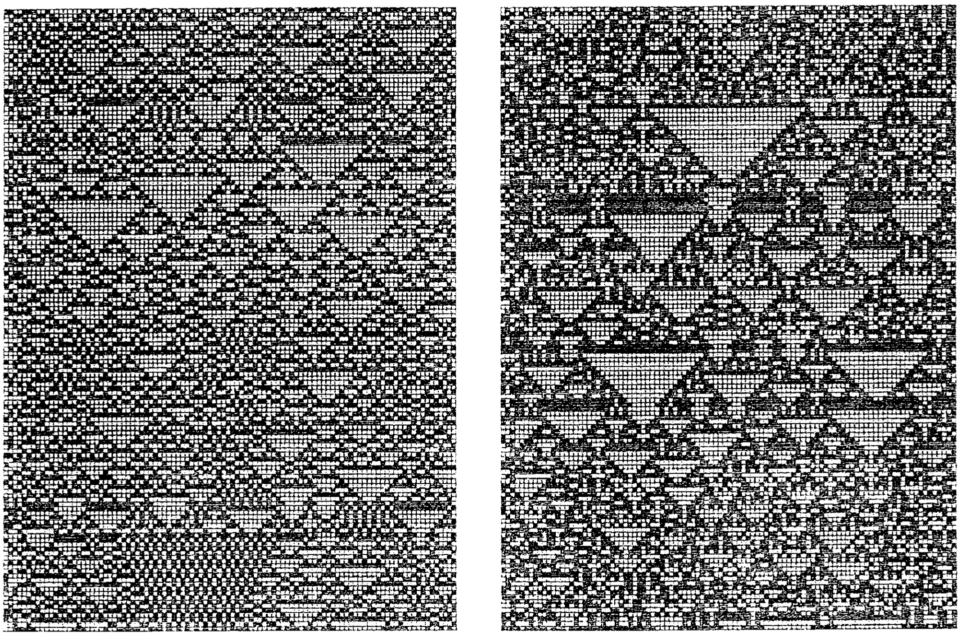


B

Fig. 8 Temporal Pattern of Interacting Ruledynamical Registers
Two ruledynamical registers are obeyed different rule change procedures. Two registers interact with the partial interchange of register element states. Each register seize is 100. The left side follows the inetermittent like ruledynamics shown in Fig. 6. The right side is the ruledynamics with parameter values, $\eta_1=-1$, $T_1=0.7$, $\eta_2=-1$, $T_2=0.8$, $\eta_3=1$, $T_3=0.6$, $\eta_4=1$, $T_4=0.4$, $\eta_5=-1$, and $T_5=0.2$. The swapping area is the left side of bit arrays from the left end. A : no swapping B : 1 bit swapping

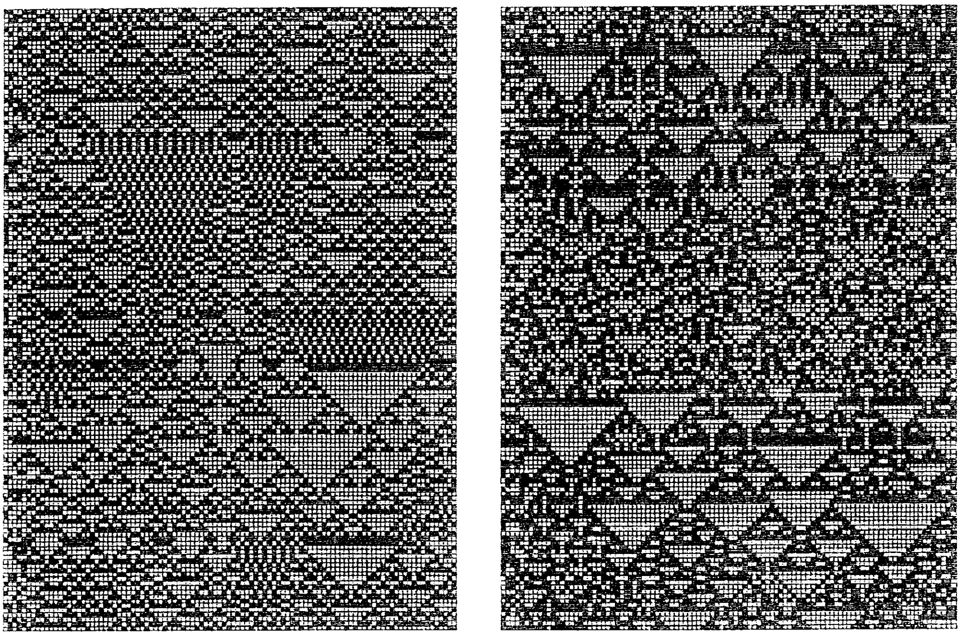


C

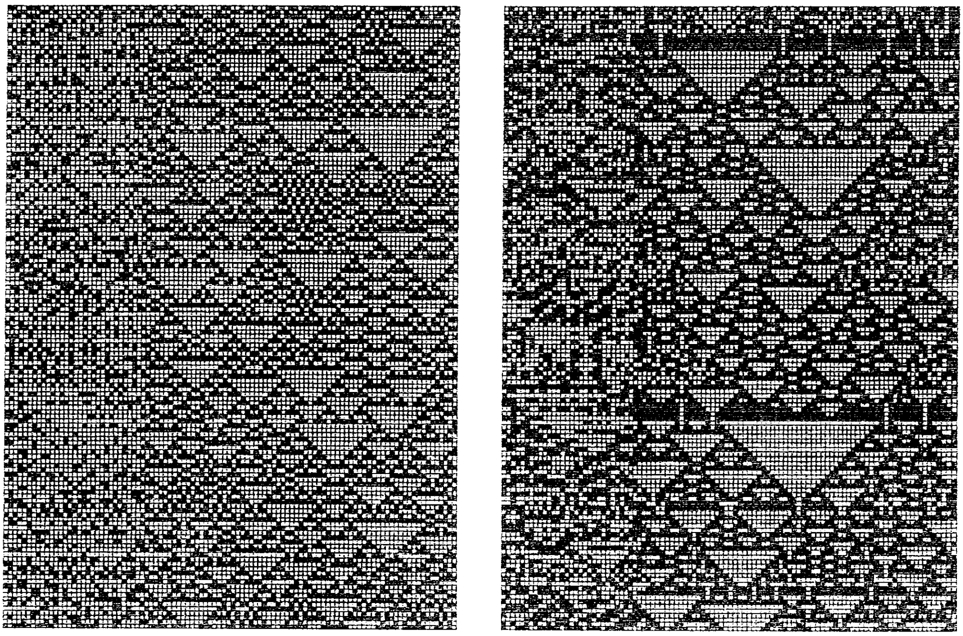


D

Fig. 8 (continued) C : 2 bits swapping D : 3 bits swapping



E



F

Fig. 8 (continued) E : 5 bits swapping F : 10 bits swapping

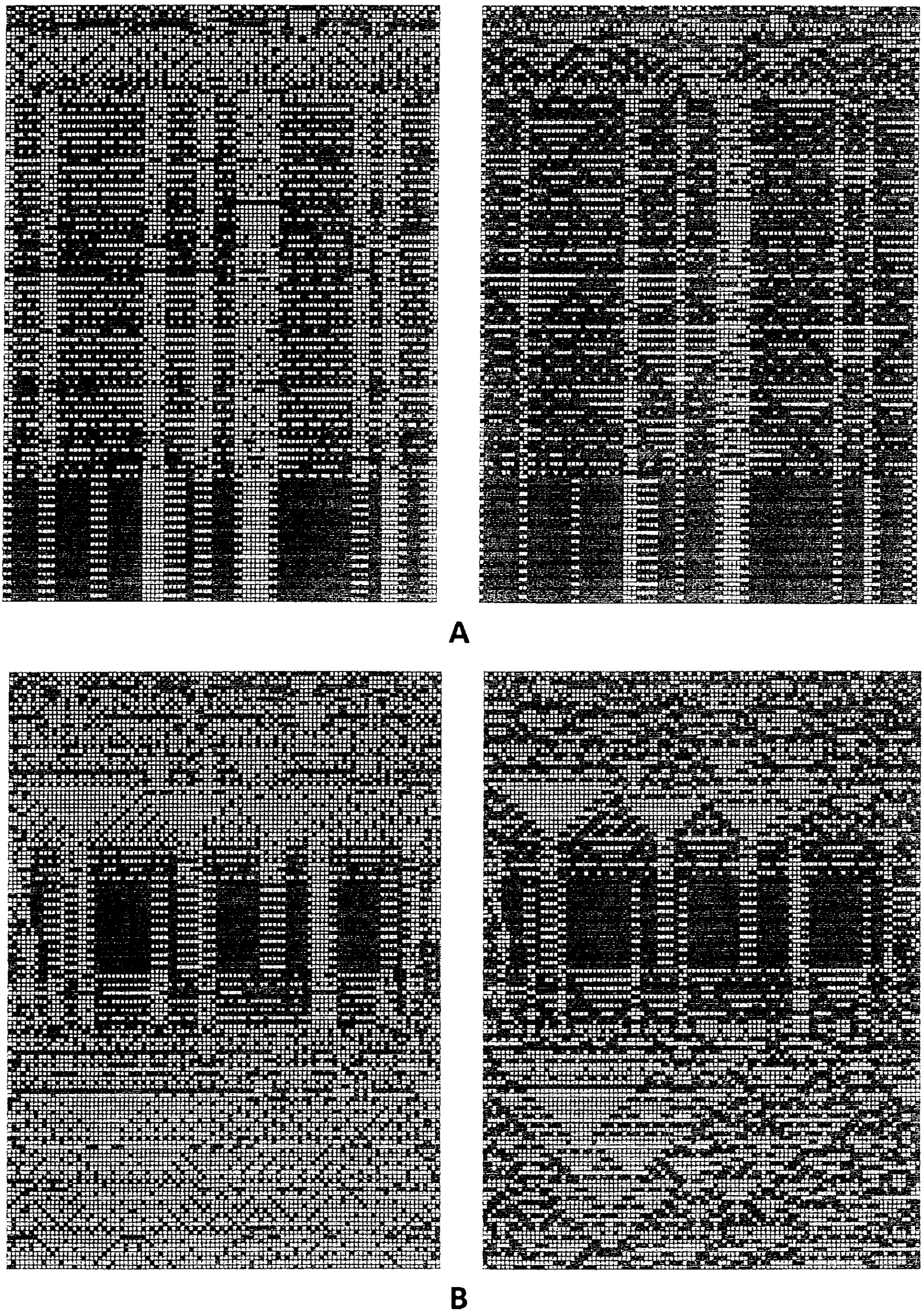
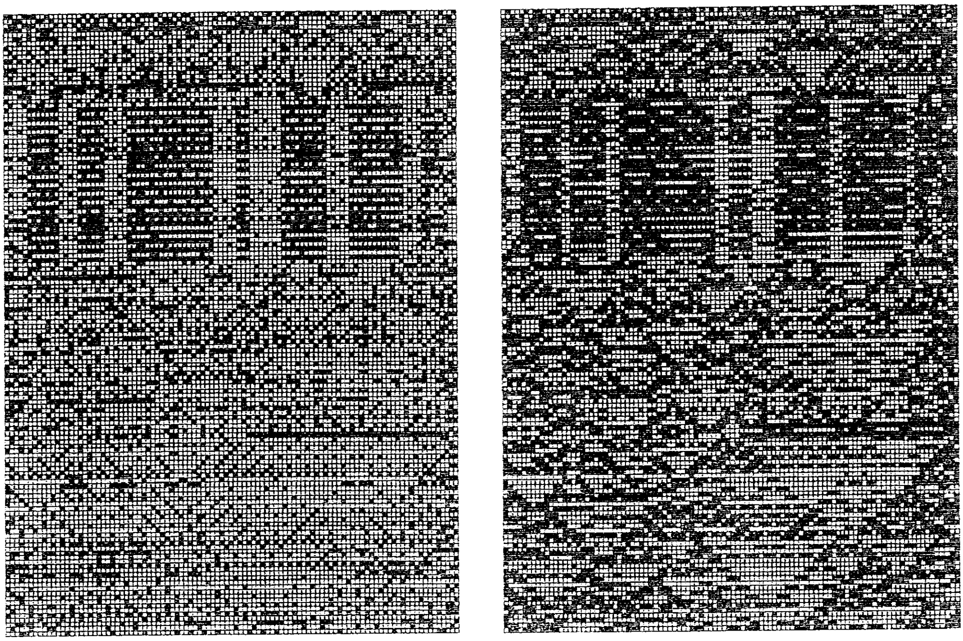
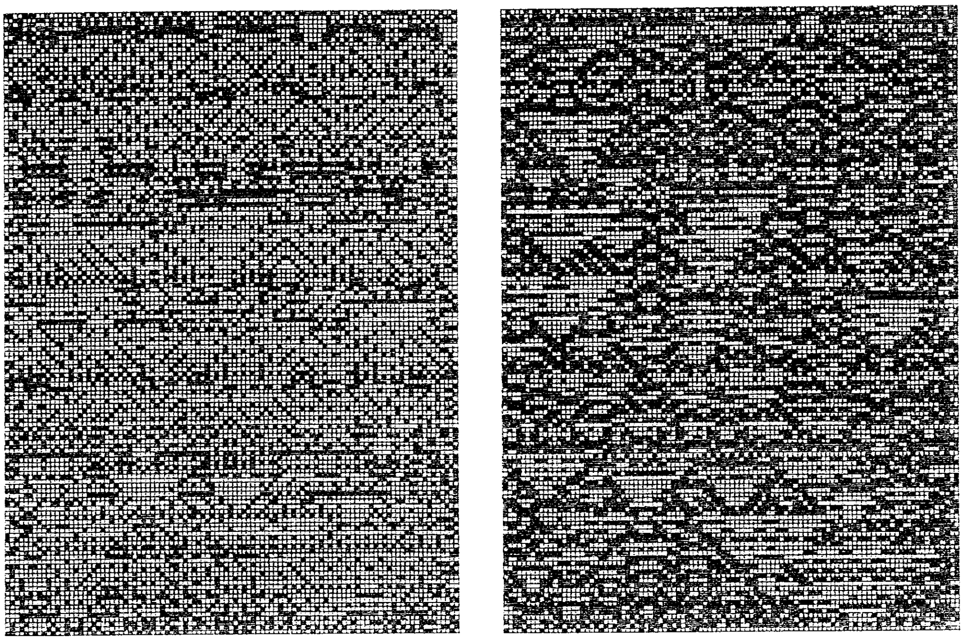


Fig. 9 Temporal Pattern for the Case of Register Bits Almost Swapping
Parameter values of ruledynamics are the same as shown in Fig. 8.
A : whole swapping B : no swapping 1 bit



C



D

Fig. 9 (continued) C : no swapping 2 bits D : no swapping 3 bits array of binary digits. Any size of swapping binary data also brings about the different type of pattern dynamics to each other. We however see the similarity of temporal pattern in fractional area of pattern dynamics. The similarity of pattern dynamics is equivalent to the average activity synchronization demonstrated by Kim and Aizawa[15]. Swapping areas of register memories perform the different pattern changes from the non-swapping area. This is easily observed when the size of swapping areas is lengthened. Based on these results we conclude that ruledynamical system has a kind of robustness or rigidity for the swapping interactions by the partial interchange of memory states. To see what happens by the swapping interaction of memory state, we investigate the pattern dynamics for the whole memory state swapping case and the cases where a few size of non-swapping memory regions remain in each interacting register. The simulation results are shown in Fig. 9. Fig. 9(A) is the case of whole memory state swapping. In this case, pattern dynamics shows synchronous behavior of registers. Two registers do not coincide completely, but pattern dynamics of these registers show the obscure synchronization. The cases of a few non-swapping memory state (Fig. 9 (B-D)) also shows us the rigidity of ruledynamics. Since ruledynamics have a kind of rigidity, the swapping part shows the obscure synchronization of pattern dynamics. This fact implies any computer system can follow other computer processes and also maintaining its identity through the proper rule change given in a ruledynamical system.

Now we formulate the system of interacting registers mathematically. Using the state vector of registers $(S_1(t), S_2(t))$, the ruledynamical system with swapping interaction of cell states can be expressed simply, namely,

$$\begin{aligned} S_1(t+1) &= \Sigma\varepsilon_\chi^1(S_1(t)-\phi_m(S_1(t))+\phi_m(S_2(t)))F_\chi(S_2(t)-\phi_m(S_1(t))+\phi_m(S_2(t))) \\ S_2(t+1) &= \Sigma\varepsilon_\chi^2(S_2(t)-\phi_m(S_2(t))+\phi_m(S_2(t)))F_\chi(S_2(t)-\phi_m(S_2(t))+\phi_m(S_1(t))) \end{aligned} \quad (3.1)$$

where ε_χ and F_χ are the same notations as in eq. (2.2), and $\phi_m(S(t))$ means the data extracted from swapping interaction area in the register. Notice that the every value of $\phi_m(S(t))$ outside of swapping interaction area is zero.

The intrinsic properties of a ruledynamical system is caused by the term $\varepsilon_\chi(S_i(t))$ $(i=1, 2)$. If this term is given through the interaction of two registers, then the expression becomes follows:

$$\begin{aligned} S_1(t+1) &= \Sigma\varepsilon_\chi^1(S_1(t), S_2(t))F_\chi(S_1(t)) \\ S_2(t+1) &= \Sigma\varepsilon_\chi^2(S_1(t), S_2(t))F_\chi(S_2(t)). \end{aligned} \quad (3.2)$$

This denotes another type of the communication between computers. As seen from these two type of equations (3.1) and (3.2), it is obvious that several kinds of computer communications are possible. We imagine that the identity of a ruledynamical system is brought about from the term of $\varepsilon^i(\cdots)$ $(i=1, 2)$ in Eq. (3.1), namely, the ruledynamical independency appears in the case where the coefficient of each rule is determined only by the intrinsic quantities of a system.

To consider the results of the case for two communicating registers shown in Figs. 8 and

9, it is appropriate to write the equations to describe the temporal development of each element. The equation for every element is given by the following expressions,

1. Non-swapping areas

1.1 Non-boundary regions

$$S_i^1(t+1)=\Sigma\theta(\eta_k^1(\langle S_i^1\rangle_t^* - T_k^1)g_k(S_{i-1}^1(t), S_i^1(t), S_{i+1}^1(t)), \bmod 2,$$
$$S_i^2(t+1)=\Sigma\theta(\eta_k^2(\langle S_i^2\rangle_t^* - T_k^2)g_k(S_{i-1}^2(t), S_i^2(t), S_{i+1}^2(t)), \bmod 2, \qquad (3.3)$$

1.2 on Boundaries

$$S_{m+1}^1(t+1)=\Sigma\theta(\eta_k^1(\langle S_i^1\rangle_t^* - T_k^1)g_k(S_m^2(t), S_{m+1}^1(t), S_{m+2}^1(t)), \bmod 2,$$
$$S_n^1(t+1)=\Sigma\theta(\eta_k^1(\langle S_i^1\rangle_t^* - T_k^1)g_k(S_{n=1}^1(t), S_n^1(t), S_1^2(t)), \bmod 2,$$
$$S_{m+1}^2(t+1)=\Sigma\theta(\eta_k^2(\langle S_i^2\rangle_t^* - T_k^2)g_k(S_m^1(t), S_{m+1}^2(t), S_{m+2}^2(t)), \bmod 2,$$
$$S_n^2(t+1)=\Sigma\theta(\eta_k^2(\langle S_i^2\rangle_t^* - T_k^2)g_k(S_{n-1}^2(t), S_n^2(t), S_1^1(t)), \bmod 2, \qquad (3.4)$$

2. Swapping areas

2.1 Non-boundary regions

$$S_j^1(t+1)=\Sigma\theta(\eta_k^1(\langle S_j^1\rangle_t^* - T_k^1)g_k(S_{j-1}^2(t), S_j^2(t), S_{j+1}^2(t)), \bmod 2,$$
$$S_j^2(t+1)=\Sigma\theta(\eta_k^2(\langle S_j^2\rangle_t^* - T_k^2)g_k(S_{j-1}^1(t), S_j^1(t), S_{j+1}^1(t)), \bmod 2, \qquad (3.5)$$

2.2 on Boundaries

$$S_1^1(t+1)=\Sigma\theta(\eta_k^1(\langle S_i^1\rangle_t^* - T_k^1)g_k(S_n^1(t), S_1^2(t), S_2^2(t)), \bmod 2,$$
$$S_m^1(t+1)=\Sigma\theta(\eta_k^1(\langle S_i^1\rangle_t^* - T_k^1)g_k(S_{m-1}^2(t), S_m^2(t), S_{m+1}^1(t)), \bmod 2,$$
$$S_1^2(t+1)=\Sigma\theta(\eta_k^2(\langle S_i^2\rangle_t^* - T_k^2)g_k(S_n^2(t), S_1^1(t), S_2^1(t)), \bmod 2,$$
$$S_m^2(t+1)=\Sigma\theta(\eta_k^2(\langle S_i^2\rangle_t^* - T_k^2)g_k(S_{m-1}^1(t), S_m^1(t), S_{m+1}^2(t)), \bmod 2, \qquad (3.6)$$

where

$$\langle S_i^1\rangle_t^* = \frac{1}{n}\left\{\sum_{i=1}^{m} S_i^2(t) + \sum_{i=m+1}^{n} S_i^1(t)\right\}, \quad \langle S_i^2\rangle_t^* = \frac{1}{n}\left\{\sum_{i=1}^{m} S_i^1(t) + \sum_{i=m+1}^{n} S_i^2(t)\right\},$$

while T_k^l and η_k^l ($l=1,2$) are the same meaning in Eq. (3.1), and m signifies the length of swapping area. As seen from Eqs. (3.3)–(3.6), the temporal development of pattern in each non-swapping area is the same as the proper one generated by the given ruledynamics , and that in every swapping area follows the respective ruledynamics by changing bit arrays, except for on the boundary sites faced on the marginal point between non-swapping and swapping areas. The mixing use of the different binary states from both ruledynamical systems appears on only the boundary sites. This means that only one bit swapping of register element states is essential in the memory swapping interactions. A set of equations (3.3)–(3.6) reveals how the simulation results shown in Figs.8 and 9 are brought about.

References

[1] Y. Aizawa and Y. Nagai, "Dynamics on Pattern and Rule—Rule Dynamics—" (in Japanese), Bussei Kenkyu **48** (1987) 316-320.

[2] Y. Aizawa and Y. Nagai, "Rule Dynamics and Fuzzy Attractor: New Approach to EEG", in Cooperative Dynamics in Complex Physical Systems (ed. H. Takayama), Springer-Verlag, Berlin, Heidelberg, pp. 269-270 (1989)

[3] Y. Nagai and Y. Aizawa, "Hierachical Structure and Ruledynamics" (in Japanese), Technical Reports on Information Processing in IEEJ, **IP-95-19** (1995) 51-59.

[4] Y. Nagai and Y. Aizawa, "Principles of Arithmetic Calculation in Ruledynamics" (in Japanese), Technical Reports on Information Processing in IEEJ, **IP-97-5** (1997) 31-34.

[5] Y. Nagai, M. Kito, and Y. Aizawa, "Compressed Informtic Telecommunications of Medical Data" (in Japanese), in Information Technologies and Their Applications in the Present Time and the Future, Thirty Fifth Anniversary Series of Seikei Gakkai Vol. 4 Chap. 4 pp. 157-211, (1997).

[6] Y. Nagai and Y. Aizawa, "Ruledynamical Interpretation of McCulloch-Pitts Neuron Networks" submitted to Special Issue of Biosystem (2000)

[7] J. von Neumann, Theory of Self-Reproduction Automata (edited and completed by A. W. Burks), University of Illinois press, 1966, see Preface by A. W. Burks.

[8] S. Wolfram, Theory and Application of Cellular Automata, World Scientific, Singapore, 1986.

[9] J. P. Hayes, Computer Architecture and Organization, 2nd ed., McGraw-Hill, New York, 1988.

[10] D. E. Comer, Internetworking with TCP/IP, Vol I, 2nd ed, Prentice Hall, Englewood Cliffs, New Jersey, 1991.

[11] D. Cassel and M. Jackson, Introdction to Computers and Information Processing, 2nd ed, Reston Publishing, Reston, Virginia, 1985.
J. Ferguson, Microprocessor Systems Engineering, Addison-Wesley Publishing, Wokingham, England, 1985.

[12] K. Tezuka, Foundation of Electric Computer (in Japanese), Shokosha, Tokyo, p5, 1978.

[13] J. E. Hopcroft and J. D. Ullman, Introduction to Automata Theory, Languages, and Computation, Addison-Wesley Publishing, Massachusetts, 1979.

[14] Y. Nagai and Y. Aizawa, "One-Dimensional Fractal Map", Mem. Kokushikan Univ. CIS **19** (1998) 21-28.

[15] S. J. Kim and Y. Aizawa, "Synchronization Phenomena in Rule Dynamical Syatems", Prog. Theor. Physics **102** (1999) 729-748.