CONSTRUCTING EVOLUTIONARY SYSTEMS ON A SIMPLE DETERMINISTIC CELLULAR AUTOMATA SPACE

Hiroki Sayama

Ph.D. Dissertation

Department of Information Science
Graduate School of Science
University of Tokyo

December 1998

ABSTRACT

We succeeded for the first time in constructing evolutionary systems on a simple 9-state 5-neighbor cellular automata (CA) space by utilizing Langton's self-reproducing loop. CA are deterministic dynamical systems capable of representing extremely complex nonlinear phenomena, where time, space and states of sites are all discrete. Many CA models of self-reproductive behavior of theoretical organisms have so far been energetically studied, but the evolutionary process of organisms driven by variation and natural selection has never been realized on CA space yet. In this dissertation, we added three improvements into Langton's loop, i.e., to realize a kind of death by introducing a new dissolving state '8' into the set of states of the CA, to enhance the adaptability (a degree of the variety of situations in which the structures in the CA space can operate regularly) of the self-reproductive mechanism described by the state-transition rules of the CA, and to modify the initial structure of the loop slightly. The experiment with this improved loop met with the intriguing result that the process of spontaneous evolution emerged in the CA space, where loops varied by direct interaction of their phenotypes, fitter individuals were naturally selected, and the whole population gradually evolved toward the fittest species. The most important contribution of this work is to show that it is possible to construct evolutionary systems on such a simple mathematical medium as a CA space by introducing the mortality of individuals, their interaction, and their robustness to variations into the model. This implies that, in the future, we will be able to create extraordinary large-scale evolutionary systems in a fine-grained superparallel machine environment by using a very simple algorithm with neither explicit management of living individuals nor generation of random numbers for stochastic mutation of genotype.

論文要旨

本研究では，Langton の自己増殖ループを利用して，9 状態 5 近傍の単純なセルラオートマタ（CA）空間上に進化系を構築することに初めて成功した．CA は，極めて複雑な非線形現象を表現する能力を有する決定性力学系で，そこでは時間・空間・各セルの状態の全てが離散的である．これまでに，理論的な生物の自己増殖挙動について多くの CA モデルが精力的に研究されてきたが，変異と自然選択によって推進される生物の進化過程を CA 空間上に実現することは，未だ達成されていない．本研究では，Langton のループに対し，CA の状態集合に新たに解消状態 '8' を導入して一種の死を実現し，状態遷移規則によって記述されている自己増殖メカニズムの融通性（CA 空間内の構造体が正常に動作する状況の多様さの程度）を高め，かつループの初期構造を若干変更するという 3 つの改良を加えた．この改良型ループを用いた実験では，自発的な進化過程が CA 空間内に創発するという興味深い結果が得られた．そこでは表現型同士の直接の相互作用によってループが変異し，より適応度の高い個体が自然選択され，個体群全体が徐々に最適な種へと進化していった．本研究の最も重要な貢献は，生命体の可死性，それらの相互作用，及び変異に対するロバスト性をモデルに導入することによって，CA 空間のような単純な数学的媒体を用いて進化系を構築することが可能であることを示した点にある．これは，生存個体の明示的な管理や遺伝型の確率的変異に用いる乱数の生成を伴わない非常に簡単なアルゴリズムを用いて，細粒度超並列計算機環境上に極めて大規模な進化系を実現することが，将来的に可能であることを示唆している．

Acknowledgments

First of all I would like to express here my great respect for Chris G. Langton who gave me many pieces of valuable advice, as well as for his works which forms the very basis of this study.

I am greatly indebted to my supervisor Prof. Yoshio Oyanagi for his appropriate guidance and generousness to let me "run loose" in such an odd research field. Discussions with Reiji Suda, Osamu Tatebe, Akira Nishida, and all colleagues in Oyanagi laboratory, were quite fruitful for me in proceeding with this work. Prof. Toshihisa Takagi, the chief examiner responsible for this dissertation, was kind enough to spend a considerable amount of time to discuss with me, which led me to a much deeper understanding. Ikuo Uchiyama in Takagi laboratory also gave me valuable comments for this study from a biological viewpoint. Wayne Dawson in Takagi laboratory took the trouble for me to review the draft of this dissertation and gave me many pieces of significant advice as well as thorough revision of my English expression, which I believe greatly contributed to the improvement in quality of this dissertation. Nobuyuki Umezu, Taku Kohmura, and Yoshihisa Shinagawa in Shinagawa laboratory played an indispensable role in visualizing the experimental results obtained in this study. I also have been greatly supported by the interest and encouragement granted to this study from the following: Yasuhiro Suzuki, Keiki Takadama, Takaya Arita, Chris Adami, Gavin E. Crooks, Martin Zwick, and many other people.

I have no words to express my gratitude to my parents and siblings: Yasuhiro, Chiaki, Yasuki, and Mami Kato, for their understanding and kind support. Finally, I would like to thank, from the bottom of my heart, my dearest wife Mari and my first son Takehiro, who always provide me with all the energy for my study and even for my life itself. I could never complete this work without their continuous support and encouragement.

Special thanks to DEC Digital Personal Workstation 500au that served as a principal platform for processing the enormous computations involved in this study, and to JCC Workstation JS5/170 that was my personal terminal machine where I could operate many simulations and typeset this dissertation comfortably.

Contents

List of Tables v

List of Figures ix

1 INTRODUCTION 1

1.1 Motivation 1

1.2 Contents of this dissertation 3

2 CELLULAR AUTOMATA 4

2.1 Overview of research on CA 4

2.2 Formal definitions 6

2.3 Examples 8

2.4 Behavioral classification 13

2.5 Practical applications 14

3 ARTIFICIAL LIFE ON CELLULAR AUTOMATA 15

3.1 Concept of artificial life 15

3.2 Studying artificial life on CA 16

3.3 Formal definitions 17

3.4 Former models of artificial life on CA 19

3.4.1 Creating a universal constructor-computer 19

3.4.2 Searching for a minimal system capable of non-trivial self-replication 21

3.4.3 Applying to practical problems 24

3.4.4 Realizing emergence and evolution 28

4 STRATEGY OF THIS STUDY 33

4.1 Target 33

4.2 Problems and basic ideas for solution 35

4.3 Distinction between two aspects of death 38

4.4 Subject matter: Langton's self-reproducing loop 39

4.5 Experimental tools used in this study 46

5 A STRUCTURALLY DISSOLVABLE SELF-REPRODUCING LOOP 47
5.1 Introduction . . . 47
5.2 Implementation . . . 47
5.2.1 Constructing state-transition rules . . . 47
5.2.2 Fundamental behaviors . . . 50
5.3 Experiments . . . 52
5.3.1 Self-reproduction in infinite space . . . 52
5.3.2 Self-reproduction in finite space . . . 52
5.3.3 Struggle for existence in finite space . . . 56

6 EVOLOOP: AN EVOLVING SDSR LOOP 59
6.1 Introduction . . . 59
6.2 Implementation . . . 60
6.2.1 Problem in the state-transition rules of the SDSR loop . . . 60
6.2.2 Reconstructing the state-transition rules . . . 61
6.2.3 Fundamental behaviors . . . 62
6.2.4 Developing a method of observation . . . 66
6.2.5 Evaluation of the adaptability . . . 69
6.2.6 Preliminary experiments . . . 70
6.2.7 Modifying the initial structure . . . 73
6.3 Experiments . . . 76
6.3.1 Self-reproduction and evolution in finite space . . . 76
6.3.2 Relation between resource size and evolvability . . . 80
6.3.3 Fault tolerance . . . 81
6.3.4 Emergence of self-reproducing organisms . . . 83
6.4 Problems . . . 85

7 DISCUSSIONS 86
7.1 Significance of structural dissolution . . . 86
7.2 Strategic death based on structural dissolution . . . 88
7.3 Evolution driven by direct interaction of phenotypes . . . 89
7.4 Characteristics as artificial life implemented on CA . . . 90
7.5 Characteristics as an artificial evolutionary system . . . 91
7.6 Evolvability of artificial evolutionary systems . . . 93

8 CONCLUSION AND FUTURE EXTENSIONS 95
8.1 Summary of this study . . . 95
8.2 Future extensions . . . 96
8.2.1 Facilitating the improvement of CA . . . 96

8.2.2 Removing restrictions on the shape of organisms 97
8.2.3 Diversifying the function of and the relation between organisms . . . 98

Appendix **100**

A STATE-TRANSITION RULES OF LANGTON'S SR LOOP **100**

B STATE-TRANSITION RULES OF THE SDSR LOOP **102**

C STATE-TRANSITION RULES OF THE EVOLOOP **103**

D BREAKDOWN OF STATE-TRANSITION RULES OF THE EVOLOOP ACCORDING TO THEIR FUNCTIONS **105**

D.1 Phase 1: To lengthen the construction arm 105
D.2 Phase 2: To turn the tip of the arm left 106
D.3 Phase 3: To bond the tip and the root of the arm together 106
D.4 Phase 4: To dissolve the umbilical cord between parent and offspring 108
D.5 Phase 5: To germinate a new sprout of the arm 110
D.6 Phase 6: To lengthen the new sprout of the arm 111
D.7 General rules for the sustenance of sheath structures and the propagation of genes 112

Bibliography **115**

List of Tables

4.1 Names and functions of the states in the CA of Langton's SR loop 40

5.1 Records of a round-robin among the SDSR loops of species from 4 to 12 . . 58

6.1 Names and functions of the states in the improved CA 63
6.2 Results of preliminary experiments of breeding *evoloops* in finite spaces . . . 72
6.3 Results of full-scale experiments of breeding *evoloops* with new genotypes . 78
6.4 Probabilities that the self-reproductive activity of *evoloops* will continue for 10000 updates with two kinds of faults provided 83

7.1 Functions of state '3' and '4' in different phases of the self-reproductive process of the *evoloop* 92

A.1 State-transition rules of Langton's SR loop 101

C.1 Principal part of the state-transition rules of the *evoloop* 104

List of Figures

2.1 Research fields relevant to CA 5
2.2 Schematic illustrations of the definitions of terms concerning CA and the principle of operation of CA 7
2.3 Examples of neighborhood templates often used for 2-dimensional CA . . . 8
2.4 Schematic illustrations of rotational symmetry in 2-dimensional CA with a von Neumann neighborhood 9
2.5 Behavior of binary CA with a von Neumann neighborhood governed by the XOR (parity) rule 10
2.6 Behavior of binary CA with a Moore neighborhood governed by the voting rule 11
2.7 Behavior of the famous binary CA "Game of Life" 12
2.8 Structure called "glider" and the NOT gate made with gliders in the universe of the Game of Life 12
2.9 Examples of the behaviors of 1-dimensional binary CA in four classes 13

3.1 Schematic illustrations of the idea of von Neumann's self-reproducing automaton model 20
3.2 Von Neumann's self-reproducing automaton implemented on a 29-state 5-neighbor CA space 22
3.3 Vitányi's sexually reproducing automata 23
3.4 Langton's self-reproducing loop 23
3.5 Self-reproductive process of Byl's small loop 24
3.6 Various kinds of unsheathed self-replicating loops discovered by Reggia et al. 25
3.7 Morita et al.'s self-reproducing worm with a shape-encoding mechanism . . 26
3.8 Tempesti's self-reproducing loop capable of additional construction 27
3.9 Perrier et al.'s self-reproducing loop capable of universal computation . . . 28
3.10 Chou et al.'s self-replicating loops solving a simple satisfiability problem . . 29
3.11 Example of Lohn et al.'s self-replicating structures governed by the state-transition rules automatically discovered by genetic algorithms 30
3.12 Chou et al.'s CA space where self-replicating loops emerge and evolve . . . 32

4.1 Metabolic flowchart of creatures that evolved in the world of *Tierra* 34
4.2 Schematic illustration of the difference between this study and other former ones with respect to which part of state-transition rules are mainly focused on 36
4.3 Oohashi et al.'s self-reproducing/decomposing automaton model 37
4.4 Initial structure of Langton's SR loop 39
4.5 Self-reproduction of the SR loop 41
4.6 Mechanism of straight growth of the construction arm of the SR loop . . . 42
4.7 Mechanism of left turning of the tip of the arm of the SR loop 42
4.8 Mechanism of bonding of the tip and the root of the arm followed by the dissolution of the umbilical cord in the SR loop 43
4.9 Mechanism of germination of a new sprout of the arm in the parent SR loop 43
4.10 Mechanism of germination of a new sprout of the arm in the offspring SR loop 44
4.11 Death as functional failure of the SR loop 44
4.12 Mechanism of the beginning of death of the SR loop 45
4.13 Growth of the colony of the SR loop 45

5.1 Behavior of dissolving state '8' 49
5.2 Only one alteration in the state-transition rules in implementing the SDSR loop 50
5.3 Death as structural dissolution of the SDSR loop 51
5.4 Partial structural dissolution of the SDSR loop 51
5.5 Temporal development of spatial distribution of SR/SDSR loops in infinite space 53
5.6 Temporal development of the number of living individuals of SR/SDSR loops in infinite space 54
5.7 Temporal development of spatial distribution of SR/SDSR loops in finite space 55
5.8 Examples of merged loops produced by direct interaction of phenotypes in the course of self-reproduction of SDSR loops 56
5.9 Examples of loops of different species 56
5.10 Struggle for existence between SDSR loops of species 4 and 6 57
5.11 Struggle for existence between SDSR loops of species 7 and 9 58

6.1 What happens if the form of the arm of the SDSR loop is altered by force during self-reproduction 60
6.2 General behaviors of six phases of the self-reproductive process of the improved loop 64

6.3 Mechanism of the phase to bond the tip and the root of the arm together in the improved loop 65
6.4 Mechanism of the phase to dissolve the umbilical cord between parent and offspring of the improved loop 65
6.5 Mechanism of the phase to germinate a new sprout of the arm of the improved loop 66
6.6 Mechanism of the phase to lengthen the new sprout of the arm of the improved loop 67
6.7 Self-reproduction of the *evoloop* of species 13 68
6.8 Takeover of the arm caused by the collision of two *evoloops* 68
6.9 Image processing by the "observer" CA for the automatic identification of living loops in the space 69
6.10 Histograms of the distribution of lifetimes of SDSR loops and *evoloops* of species 6 71
6.11 Evidences of the evolvability of *evoloops* observed in the preliminary experiments 73
6.12 Averages of the numbers of active sites contained in the space and standard deviations of their fluctuation 74
6.13 Manner of utilization of a gene sequence in the loop 75
6.14 New genotypes of *evoloops* of species 13 which have stronger self-reproductive ability 76
6.15 Temporal development of configuration in the evolutionary process of 2-*evoloops* 77
6.16 Examples of variants produced through the repetition of self-reproduction and structural dissolution of *evoloops* 78
6.17 Example of variation of *evoloops* occurring through the direct interaction of phenotypes 79
6.18 Temporal development of numbers of living *evoloops* and their genealogy . . 80
6.19 Temporal development of the diversity of *evoloops* in several cases with different sizes of space 81
6.20 Snapshot taken in the evolutionary process of *evoloops* in the space of 1500 × 1500 sites 82
6.21 State-transition diagram of the stochastic device embedded in the space . . 84
6.22 Example of the emergence of self-reproducing organisms from empty space . 84

7.1 Analogy between the development of digital organisms made by computer programs and the development of self-reproducing loops on CA 93

8.1 Image of unsheathed *evoloops* on CA with weak rotational symmetry 97

8.2 Image of organisms functionally interacting with each other through the "memory" CA 99

Chapter 1

INTRODUCTION

> *... If we could populate a large area with multiple copies of such reproducing colonies, and introduce variation into at least the portion of the description that codes for the extra machinery, we would have all of the raw material necessary for natural selection to operate among variants and hence we would have a sufficient basis for the process of evolution.*
>
> — *C. G. Langton, Studying artificial life with cellular automata* [28]

1.1 Motivation

Our work presented here is a challenge to take a slight but positive, promising step forward in synthesizing living systems artificially in order to understand the substance of life more deeply.

The desire to understand the principles of life seems to be inherent in human nature. Ultimately, it may be connected with the desire to know what kind of beings we are. However, living systems including ourselves show very complex behaviors which are far from deducible and which conservative natural sciences have not been able to address successfully. Although they completely obey the fundamental laws of physics and chemistry in micro level, they often display logical, adaptive, autonomous, strategic, and even evolutionary behaviors in macro level. Thus, to define and describe living systems formally is an extremely difficult task.

The traditional approaches to the living systems employed until the middle of the twentieth century could not escape from being either mechanistic or taxonomic. Fortunately, the rapid progress of the theory and technology of computers in the last several decades has enabled us to take a new *synthetic* approach to living systems, in which we try to understand the behavior of life through constructing and simulating models of artificial living systems. This approach was recently named "artificial life" by Langton[29], but its

origin can be derived from theoretical works in 1940's much related to the beginning of computer sciences, as remarked later. In this discipline, a number of lifelike behaviors have been examined by various means such as computer simulation, neural networks, mechanical engineering, biochemical engineering, and so forth. Several well-designed systems comprised of digital organisms have been proposed so far in artificial life research[1, 37, 41], which, in a sense, possess the same properties as real life. The evolutionary process has been realized already in these systems.

However, there are still many gaps between digital organisms in these systems and real carbon-based organisms with respect of complexity and diversity. Many kinds of causes to generate these gaps can be considered, such as lack of degree of freedom, insufficiency of scale, lack of the ability to form hierarchical organizations, for example. Among them, we believe that a great leap in expanding simulation scale is one of the most important factors for the reduction of such gaps between artificial and real life. While even a coliform bacillus consists of tens of millions of biochemical macromolecules and ions[20], the digital organisms in the above systems consist of tens of thousands of instructions or memory cells at most. Such a quantitative difference is likely to produce a qualitative difference through a hierarchical emergent process.

However, it is difficult for the artificial systems mentioned above to enlarge simulation scale linearly to incomparably larger size (say, one thousand times as large as before), because such systems generally involve rather complicated computational processing for the explicit simulation of tasks such as management of living or dead individuals, introduction of mutations, and so on.

Our motivation of this study is to solve this problem, i.e., to frame the basis for the realization of extremely large-scale evolutionary systems. For this purpose, we have attempted to construct them in a simple cellular automata (CA) space which is a fine-grained computational model intrinsically suitable for the linear enlargement of simulation scale, especially by implementing it on hardware modules[51, 55]. Thus, if we could create artificial evolutionary systems on CA, it would be very significant for the rapid enhancement of simulation scale in the near future.

This dissertation gives an affirmative answer to the question whether it is possible to construct an *evolutionary process*—here we view this phrase as a process in which self-replicators vary and fitter individuals are naturally selected to proliferate in the colony—by utilizing and tuning up a simple deterministic CA space. In this study, we added three improvements into Langton's self-reproducing loop, i.e., to realize a kind of death by introducing a new dissolving state '8' into the set of states of the CA, to enhance the adaptability (a degree of the variety of situations in which the structures in the CA space can retain their regular operations) of the self-reproductive mechanism described by the state-transition rules of the CA, and to modify the initial structure of the loop slightly.

The experiment with this improved loop met with the intriguing result that the process of spontaneous evolution emerged in the CA space. It is suggested in this work that introducing the mortality of virtual organisms, their interaction, and their robustness to variations into the model is quite significant in making artificial systems capable of evolution.

We believe that this study certainly enhances the step toward the representation of the process of natural selection and evolution by means of a set of interacting *virtual state machines* embedded in CA, as foretold by Langton[28].

1.2 Contents of this dissertation

This dissertation contains eight chapters including this introductory part. The following two chapters 2 and 3 are alloted for providing readers with background knowledge useful for comprehending the contents of this study. In chapter 4 the strategy we employed to create evolutionary systems on CA is remarked as well as the introduction of Langton's self-reproducing loop which we adopted as a subject matter of this dissertation. Chapter 5 and 6 are the main parts of this dissertation, where we give a detailed account of our new models, the structurally dissolvable self-reproducing (SDSR) loop[44, 47] and the *evoloop*—an evolving SDSR loop[45, 46, 48], in terms of implementing methods and experimental results. Several issues on these two models are discussed in chapter 7 from both biological and engineering viewpoints, followed by concluding chapter 8 with several ideas of future extensions of the model. The complete state-transition rules of the models reported in this dissertation are carried in appendices.

Since the research of artificial life (especially that implemented on CA) is not so popular, we intentionally spared many pages for the description of background knowledge and related works in this dissertation. In addition, a number of figures are inserted for the depiction of the behavior of CA which is generally difficult to be described in words because of its complexity. We hope these will help readers well to understand the contents of this study.

Several color movies of the loops in action are on view at the author's WWW page:

`http://proton.is.s.u-tokyo.ac.jp/~sayama/sdsr/`

They also may be useful for the readers in understanding the behaviors of the SDSR loops and the *evoloops* introduced in this dissertation.

Chapter 2

CELLULAR AUTOMATA

Our work reported in this dissertation is a kind of study on artificial life implemented on cellular automata (CA). The ideas of both artificial life and CA have been strongly related with each other. In this chapter, the fundamental knowledge about CA necessary for understanding the contents of this study, covering related fields, definitions, and terminology, is provided to the readers.

2.1 Overview of research on CA

CA are discrete dynamical systems comprised of uniform array of identical automata and their local interaction. Since their formal definitions are provided in the next section, only a rough overview of research on CA is introduced first. Figure 2.1 shows the bird's-eye view of research fields relevant to CA, in which we shall grasp a historical position of this study.

The idea of CA was invented in the 1950's by von Neumann and his collaborator Ulam. Von Neumann was one of the principal founders in the development of modern mathematics and computer science. His aim in contriving CA was to embody the Turing machine[24] (a theoretical machine capable of any kind of computation) as a spatial pattern of states arranged in that space; thus, the birth of CA was related much deeply to the study on logic, algorithm and computation, which formed the very fundamentals of computer science.

For these several decades, however, CA have grown to be a new technique useful for modeling various kinds of complex nonlinear phenomena in computational physics and many other disciplines, because the relation between CA and traditional differential equations has been revealed[61] and several practical versions of CA have been exploited[14, 22] (Fig. 2.1: Left). The main part of recent uses of CA is of this kind—"research on physical CA", so to speak, where CA are used as a method for bottom-up modeling of various field phenomena difficult to formulate by using partial differential equations.

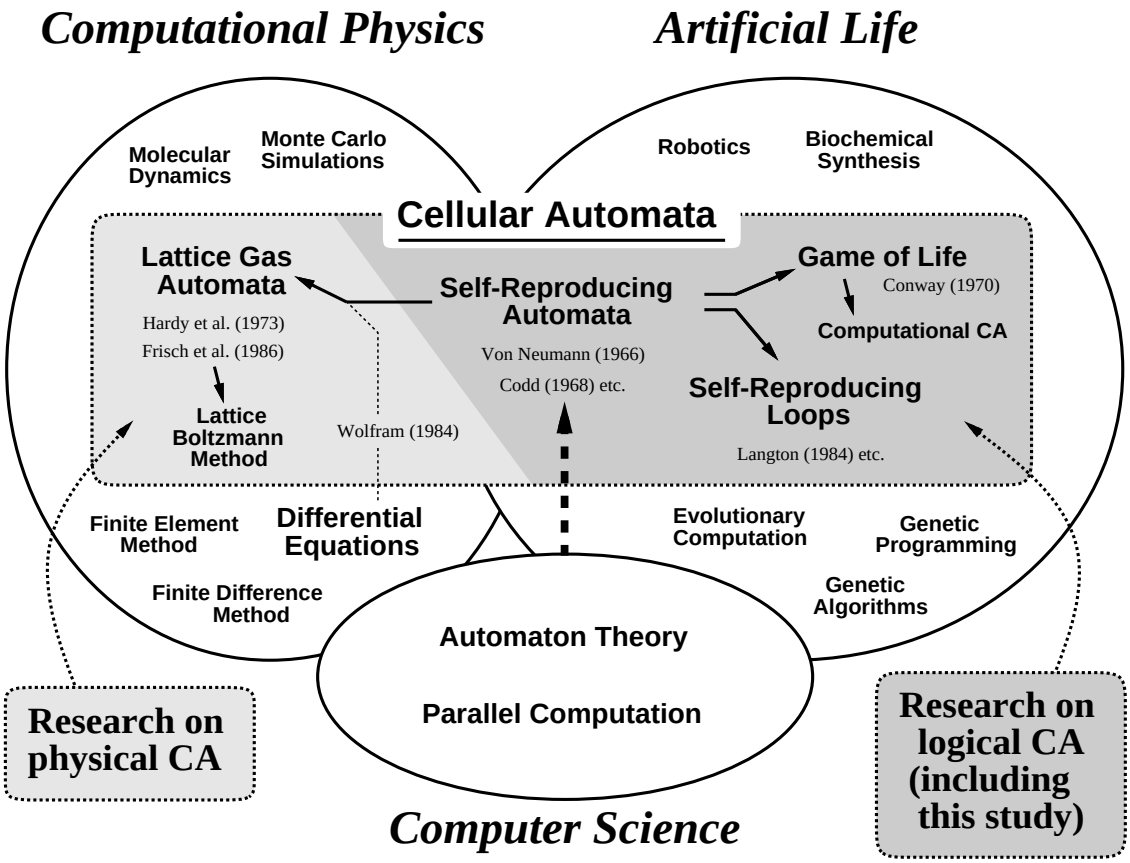


Figure 2.1: Research fields relevant to CA. In the research on physical CA (left), CA are used as a method for bottom-up modeling of various field phenomena difficult to formulate by using partial differential equations. On the other hand, in the research on logical CA (right), they are utilized as logical devices for the realization of logical structures of living systems and other computational entities. Our study presented in this dissertation also lies in the latter school.

Still, we should note that von Neumann's original goal in contriving CA was to model self-reproductive and evolvable behavior of living systems in a logical way. This attempt has recently been concerned with the very origin of artificial life—the trials to synthesize living systems by artificial means. After von Neumann's work, many kinds of efforts to create artificial organisms on CA have been continuing until this day (Fig. 2.1: Right). In these studies, being converse to the research on physical CA mentioned above, CA are used as logical devices for the realization of logical structures of living systems and other computational entities, thus we shall temporarily call them "research on logical CA" here. Our study also lies in the forefront of this school.

2.2 Formal definitions

CA are dynamical systems capable of representing extremely complex nonlinear phenomena, where time, space and states of sites are all discrete. It consists of identical automata (cells or sites) uniformly arranged on the lattice points of a D-dimensional discrete space Z^D, whose temporal development is formally represented by

$$s_{t+1}(\boldsymbol{z}) = \Delta(s_t(\boldsymbol{z}+\boldsymbol{z}_0), s_t(\boldsymbol{z}+\boldsymbol{z}_1), \ldots, s_t(\boldsymbol{z}+\boldsymbol{z}_{n-1}))$$

where $s_t(\boldsymbol{z}) \in \Sigma$ is the state of an automaton located in coordinate $\boldsymbol{z} \in Z^D$ at time t, Σ is the set of finite states each automaton may take, $\Delta : \Sigma^n \rightarrow \Sigma$ is the *state-transition rules*, $N = (\boldsymbol{z}_0, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_{n-1})$ $(\boldsymbol{z}_i \in Z^D)$ is the *neighborhood template*, respectively. s_t, which is considered to be a function mapping coordinates to states, is referred to as a *configuration* of the CA at time t. A configuration intuitively means the spatial pattern that the CA display at that time. These definitions are illustrated in Figure 2.2.

Usually $\boldsymbol{z}_0 = \boldsymbol{0}$, and $|\boldsymbol{z}_i - \boldsymbol{z}_0| \leq r$ $(i = 1, 2, \ldots, n-1)$ where r is a constant value representing the *radius* of N; namely, the state that each site will take after one update is determined locally according to the states it and other sites in its vicinity have at present. This is the reason why N is called a "neighborhood" template. A specific arrangement of states on the sites in a neighborhood template is referred to as a *situation* in this dissertation.

The transition of states occurs simultaneously and uniformly to all the sites in the CA space. The principle of such operation of CA is also shown in Figure 2.2. It is intuitively understood from this illustration that the CA is a model that represents the temporal development of a spatial pattern by only the local interactions among cellular sites which are governed by a given set of state-transition rules. These rules can be given in a form of a look-up table as shown in Figure 2.2; they also can be described by some mathematical formula or a more high-level language.

CA with $D = 1, 2, 3$ have been practically studied so far. Figure 2.3 represents typical

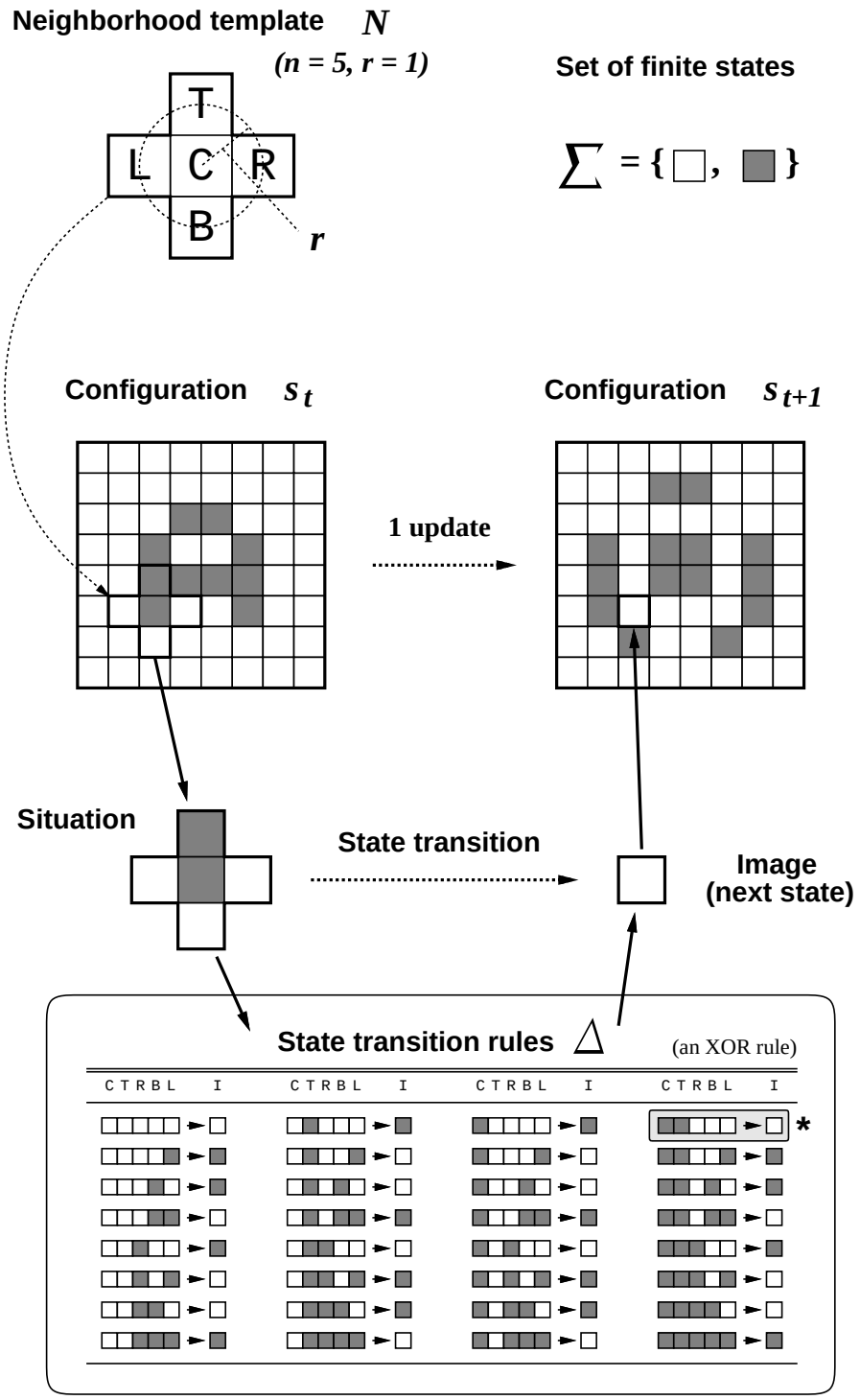


Figure 2.2: Schematic illustrations of the definitions of terms concerning CA and the principle of operation of CA. The example shown here is a 2-dimensional binary (two-state) CA space with a von Neumann neighborhood (see Fig. 2.3) which is governed by an XOR rule (see section 2.3). The next state that each site will take after one update is locally determined to be the image to which the state-transition rules map from the situation in its vicinity (designated by the asterisk in this figure). This manner of state transition occurs simultaneously and uniformly to all the sites in the CA space.

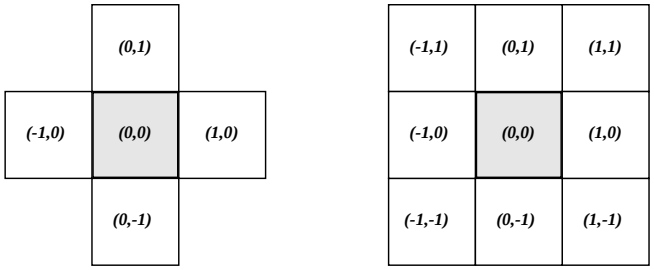


Figure 2.3: Examples of neighborhood templates often used for 2-dimensional CA. $n = 5$ in the von Neumann neighborhood, and $n = 9$ in the Moore neighborhood.

examples of neighborhood templates often used for 2-dimensional ($D = 2$) CA. In CA that adopt the von Neumann neighborhood (Fig. 2.3: Left), each site will change its state according to the states of upper, lower, right and left neighbor sites as well as itself, so $n = 5$. With the Moore neighborhood (Fig. 2.3: Right), the diagonal sites are added to the template, thus $n = 9$.

If the state-transition rules always assign the (rotationally) same state to all the situations which can be identical to each other when rotated, then that CA space is regarded as being equipped with *rotational symmetry*. It is often employed by the CA to simulate physical phenomena occurring in the rotationally symmetric universe. The rotational symmetry is called *strong* if all the states of the CA are not oriented and if rotating operation of every situation involves no replacement of states (Fig. 2.4: Left). Otherwise, it is called *weak* (Fig. 2.4: Right). In the CA with weak rotational symmetry, some states are oriented, and rotating operation in such CA involves alteration of the orientation of these states besides alteration of their coordinates.

The states of CA are usually classified as either *quiescent* or *non-quiescent*. The quiescent state remains as is when all the sites in its neighborhood are in the same quiescent state. Most CA have one basic quiescent state, often represented by either '0' or ' ' (blank), which symbolizes a vacuum in that universe. All other non-quiescent states are also called *active*, which interact with each other as well as with the quiescent state and usually play a main part in producing complex behaviors of that CA.

2.3 Examples

Several simple examples of 2-dimensional binary (two-state) CA are shown here to assist the readers' comprehension.

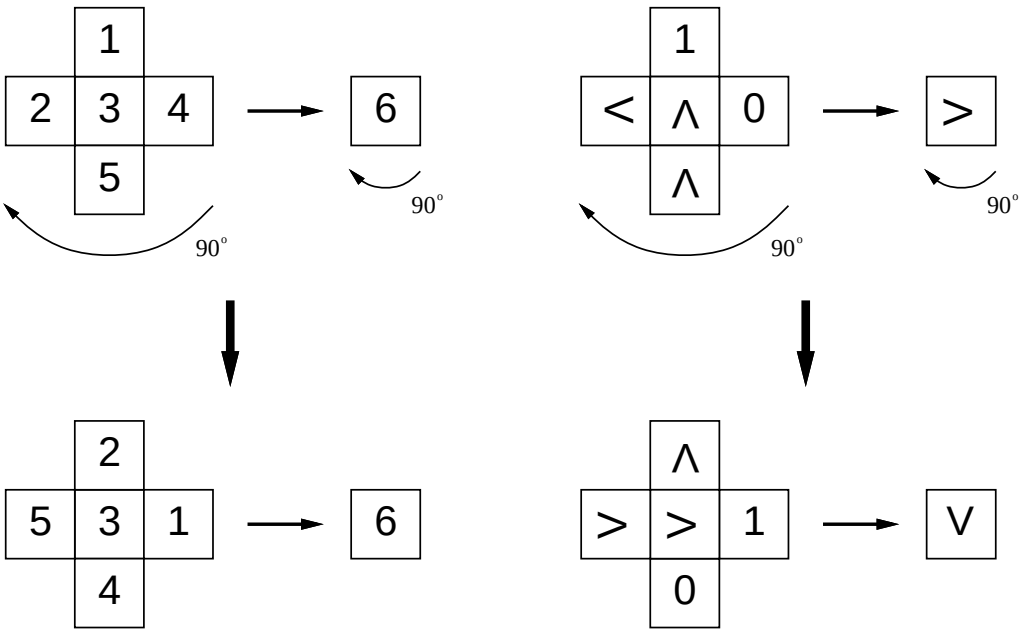


Figure 2.4: Schematic illustrations of rotational symmetry in 2-dimensional CA with a von Neumann neighborhood. In the CA with strong rotational symmetry (left), all the states are not oriented, and rotating operation involves no replacement of states. In the CA with weak one (right), rotating operation involves alteration of the orientation of some states besides alteration of their coordinates.

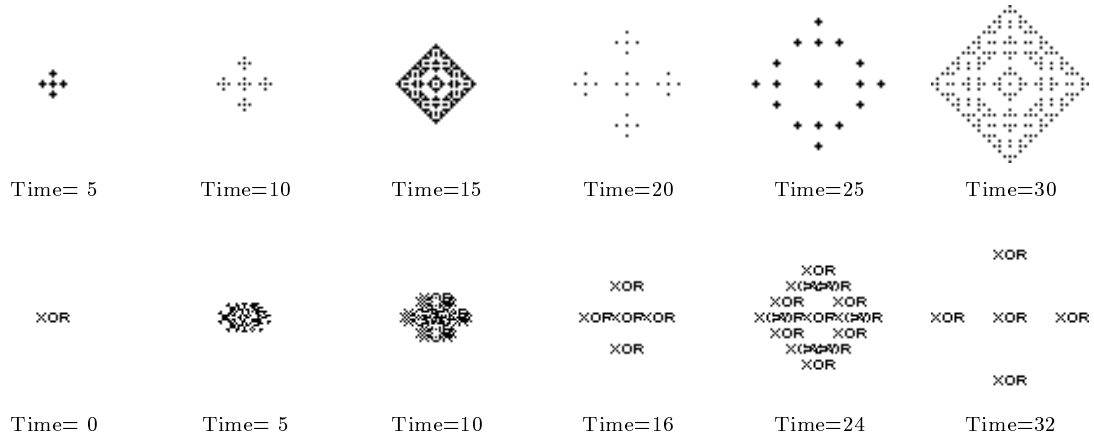


Figure 2.5: Behavior of binary CA with a von Neumann neighborhood governed by the XOR (parity) rule. Black squares represent active states, and blank space is filled with quiescent states. Upper: Development of configuration from only one active site. Lower: Self-replication of an initial structure. In this rule, any arbitrary structure can replicate itself.

XOR rule

Figure 2.5 shows the behavior of a CA space governed by an XOR rule with a von Neumann neighborhood, where the next state of each site is the result of exclusive OR of all the states in its neighborhood. This rule can be translated for binary CA as that the next state of each site becomes active if the number of active states in its neighborhood is odd, otherwise it becomes quiescent; so it is also called the parity rule[55]. This rule can be formulated as

$$\begin{aligned} s_{t+1}(\boldsymbol{z}) &= \bigoplus_{i=0}^{n-1} s_t(\boldsymbol{z}+\boldsymbol{z}_i) \\ &= \sum_{i=0}^{n-1} s_t(\boldsymbol{z}+\boldsymbol{z}_i) \pmod 2 \quad \text{(in binary CA)}, \end{aligned}$$

where $n = 5$, $\boldsymbol{z}_0 = (0,0)$, $\boldsymbol{z}_1 = (0,1)$, $\boldsymbol{z}_2 = (1,0)$, $\boldsymbol{z}_3 = (0,-1)$, $\boldsymbol{z}_4 = (-1,0)$, for example in this case. In this universe, even one active site can display rather complex growth, and furthermore, any arbitrary structure can replicate itself and propagate over the space indefinitely due to the linearity of the rule[55]. This feature is universal for all CA with XOR rules regardless of the numbers of either states or neighborhood sites.

Voting rule

The behavior of the next CA shown in Figure 2.6 may be more interesting than that of the above. This CA space is with a Moore neighborhood and governed by a voting rule[55].

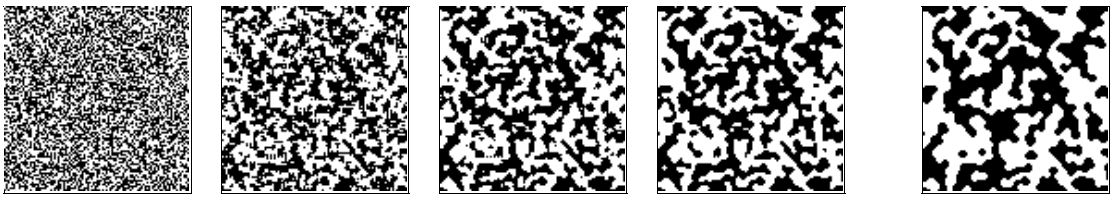


Figure 2.6: Behavior of binary CA with a Moore neighborhood governed by the voting rule. The space is of 100×100 sites with periodic boundary conditions. The initial configuration is randomly filled with quiescent and active states of the same probabilities. This CA rapidly self-organizes to the fixed pattern which looks like a phase separation of two liquids immiscible with each other.

In this rule, the next state of each site is decided by a majority vote among the states in its neighborhood; in other words, the next state of that site becomes the one that more sites take than the other in its neighborhood. When started from a random configuration, this CA rapidly self-organizes to a fixed pattern which looks like a phase separation of two liquids immiscible with each other.

Game of Life

The last example is the most popular binary CA with a Moore neighborhood, named "Game of Life", contrived by Conway[16, 17]. When a major scientific journal *Scientific American* carried the introductory articles about it in 1970, the Game of Life became a craze all over the world, and its fantastic behavior was energetically studied by many enthusiasts of this game. Its state-transition rules are very simple and a little analogous to birth and death of life, as follows:

- A dead (quiescent) site will turn into a living (active) site if and only if it is surrounded by exactly three living sites.
- A living site will remain alive if and only if it is surrounded by two or three other living sites. Otherwise the site will die.

The Game of Life shows quite interesting behavior (Fig. 2.7). A lot of characteristics of this game were found and examined during that craze, such as its computational power, its physical natures, the possibility of emergence of self-replicative creatures in it, and so on[4, 5, 15, 21, 40]. For example, it was proved that any kind of computation was possible within the universe of the Game of Life by constructing a universal computer with interacting "gliders" (Fig. 2.8)[5].

Important knowledge of general CA was also stored up through the activities of studying the Game of Life. It would be no exaggeration to say that the Game of Life played a

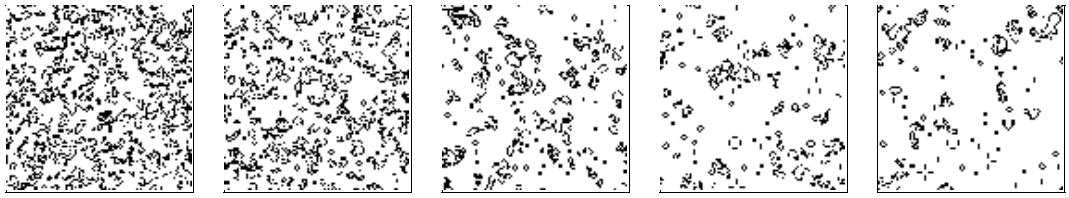


Figure 2.7: Behavior of the famous binary CA "Game of Life". The space is of 100×100 sites with periodic boundary conditions. The initial configuration is identical to that used in the previous voting rule's case. The CA shows very complex behavior too difficult to express with simple words.

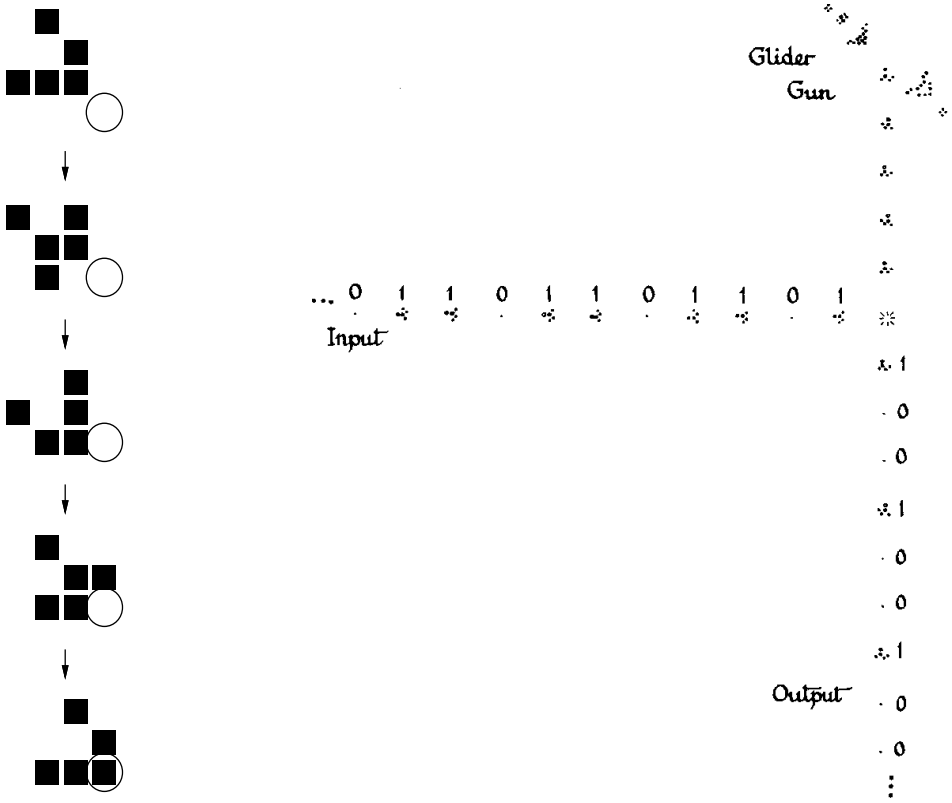


Figure 2.8: Structure called "glider" and the NOT gate made with gliders in the universe of the Game of Life. Left: Propagation of a glider. Circular marks indicate an identical site in the space. This structure moves diagonally by one site in every four updates, and travels forever if there is nothing in the way. Right: The NOT gate made with interacting gliders and the "glider gun" which ejects gliders periodically (from [5]). The input/output signals for this circuit are represented by the stream of gliders. The input stream coming from the left hand is inverted through the collision with the other stream generated by the glider gun; consequently, the result of NOT operation on the input stream goes outward as an output. In such a way, any kind of computation can be realized in the space of the Game of Life.

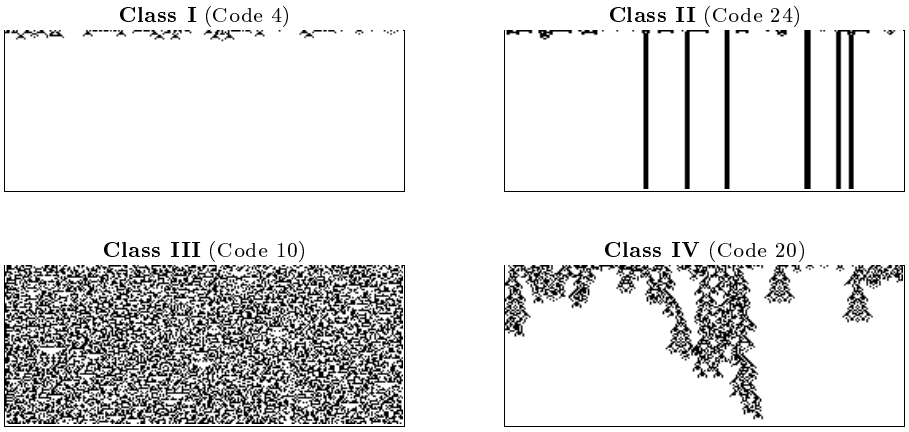


Figure 2.9: Examples of the behaviors of 1-dimensional binary CA in four classes. Here $r = 2$, thus the next state of each site after one update is affected by the states of its nearest and next-nearest neighbors as well as its own state. The space is of 250 sites with periodic boundary conditions. The initial configuration is randomly filled with quiescent and active states of the same probabilities. The initial configuration in each case is indicated under the top line of the rectangle along a horizontal axis, and its temporal development is shown downward along a vertical axis. "Code" indicates which kind of "totalistic" rules is adopted in each case (see [61] for more information about totalistic rules and their codes). CA in class I rapidly develop to a homogeneous (empty) configuration. CA in class II develop to a static but a little heterogeneous configuration. CA in class III generate quite chaotic configurations. The last CA in class IV show characteristic behavior in which several localized structures are produced.

great role in promoting the research on CA.

2.4 Behavioral classification

As reviewed above, CA generally display a great variety of development in spite of the simple principle by which they operate. The general survey of behaviors of CA was done by Wolfram[61], in which he classified the behavior of CA into the following four classes:

Class I Configuration develops to a homogeneous state.

Class II Configuration develops to a set of separated simple stable or periodic structures.

Class III Configuration develops to a chaotic pattern.

Class IV Configuration develops to complex localized structures, sometimes long-lived.

Typical examples of these behaviors of 1-dimensional binary CA are shown in Figure 2.9.

Through this classification Wolfram pointed out, for the first time, the correspondence of the behaviors of CA to those of continuous dynamical systems described by differential

equations, where class I behavior of CA is considered to correspond to fixed points in the phase space of continuous dynamical systems; similarly, class II to limit cycles, and class III to chaotic strange attractors, respectively. This discovery considerably promoted the research on CA for the modeling of physical phenomena.

According to the above classification, Wolfram predicted that the class IV behavior would be unique to discrete systems like CA and would be equipped with computational universality, i.e., the ability to execute any computational operations[24]. This regime have been actively studied from a viewpoint of informatics. For example, Berlekamp et al. proved that the Game of Life, which is the representative of the CA in class IV, actually has computational universality[5], as mentioned above. Since the computational universality involves undecidability of the halting problem of computation[24], it is believed that there is no algorithm for predicting the behavior of CA in class IV more effectively than an explicit simulation of every update.

2.5 Practical applications

Despite their capability of representing various complex nonlinear phenomena, CA are quite tractable in implementation, simulation and analysis, mainly due to the characteristics of their own, such as

1. the discreteness of space, time and states of sites,
2. the uniformity of space where an identical state-transition rule set governs the behavior of all sites in the same way, and
3. the simplicity of the state-transition rules which can be realized by using a simple look-up table.

For these reasons, CA have recently been utilized in various fields as a useful technique for modeling. They are more powerful for modeling of self-organizing field phenomena in a heterogeneous environment with complex boundary conditions, than partial differential equations, since CA can simulate the global behavior of a system by only calculating local interactions of micro-level elements.

The applications of CA range over various subjects, such as hydrodynamics, physical oceanography, material analysis, image processing, simulation of reaction-diffusion chemical systems, simulation of growth and pattern formation of living systems, ecological dynamics, traffic analysis, social and economical dynamics, and so forth. Concrete examples of these applications are not introduced here because they deviate from the contents of this dissertation.

Several technical books and articles[6, 11, 25, 50, 55, 61] are recommended for more general or detailed knowledge about CA and its practical applications.

Chapter 3

ARTIFICIAL LIFE ON CELLULAR AUTOMATA

In this chapter, we introduce the basic concept of a new scientific field "artificial life" and several former works of artificial life created by using CA.

3.1 Concept of artificial life

Artificial life is an emergent discipline of science advocated by Langton in 1987[29]. The objective of this discipline is, in short, to create living systems artificially, which aims at being the supplement of traditional biology by its new *synthetic* methodology. Various attempts to synthesize living systems are interdisciplinarily unified for this purpose, such as theoretical models, software on computers, hardware realized by electronic engineering and robotics, biochemical synthesis of microscopic structures, and so on.

In this discipline, life is considered as a form of organization independent of what kind of materials it consists of. With this viewpoint, it should be possible to liberate biology from the restriction on carbon-based life; namely, we could distinguish the essential natures of life itself from the material-dependent features of terrestrial life, and also could expand the subject matter of biology from *life-as-we-know-it* to *life-as-it-could-be*[29]. The synthetic approach to embody new virtual creatures in a virtual world and investigate them as experimental subjects is indispensable for the research on *life-as-it-could-be*. Since the essence of life is often thought of as informatical one, computer simulation of artificial organisms is quite suitable for this purpose.

However, the research of artificial life contains a serious problem that there is yet no definition of what life is. We still have to judge whether an artificial system is alive or not only in a highly subjective way. This may mean that this discipline would not have grown yet to truly scientific one in a strict sense. Conversely, however, one may say that the

activity of research on artificial life would be necessary for deepening discussions on the essence of life, because it seems that we have no choice in clarifying what we feel *lifelike* better than the repetition of a proposal of some model of life, the evaluation and criticism of that model, the proposal of another model improved according to the given criticism, the evaluation and criticism of it again, etc. To promote this process should be the main raison d'être of the research on artificial life.

Practically, the main focus of artificial life has gradually shifted these days to the applications for robotics, computer arts, and biological simulations of real creatures whose genomes have been analyzed completely. However, the primary objective of this discipline to synthesize living systems artificially seems to remain as is without any noticeable progress. The latter is the very target that this study is aiming at; namely, an orthodox challenge to create life by artificial means which was heralded in by von Neumann and established by Langton.

3.2 Studying artificial life on CA

For artificial life research including this study, CA are often utilized as logical devices to embody the mechanism of *virtual state machines*[28] imitating the behavior of life. A virtual state machine generally means a virtual automaton embedded in the space containing data operated by the actual machine. As for research on CA, the virtual state machine corresponds to some structure of the states in the CA space, while the actual machine to both the state-transition rules of CA and the external simulator for them.

The method to study artificial life by using CA was formerly theoretical, and is now based on computer simulations in most cases. Our study presented in this dissertation is also conducted by computer simulations. The practical work involved in this kind of studies is to invent somehow or other (sometimes by trial and error) both state-transition rules and an initial configuration of the CA, which should enable an objective virtual state machine to operate in the CA correctly according to the designer's intention.

The principal subjects having been modeled in these studies are the behaviors of *self-reproduction/replication* and *evolution* performed by the population of artificial organisms (actually virtual state machines). Self-reproduction/replication in a CA space is generally regarded as a phenomenon where some spatial structure (mostly an initial structure) develops in the space to its multiple copies isolated from each other after some updates. If some alteration of the structure occurs in such a copying process, it can be regarded as evolution.

It would be worth doing here to clarify the difference between self-reproduction and self-replication, since these terms have often been used in a confusing way. The difference between them has been discussed by Sipper et al.[52] According to their discussion,

replication should be regarded as an ontogenetic, developmental process which involves no operation on genomes and results in an exact copying of the parent organism. On the other hand, reproduction should be regarded as a phylogenetic, evolutionary process which involves some operations on genomes such as crossover and mutation. Therefore, most of the CA models reviewed in the following sections should be considered as self-replication rather than self-reproduction because they display merely an exact copying of artificial organisms in the CA space[1]. However, in this dissertation we tentatively conform to the terminology used in each original article.

In most CA models for self-reproduction/replication, such as von Neumann's automaton and Langton's loop introduced later, an obvious sequence of signal states which direct the self-reproduction of the organism is explicitly contained in the whole structure of the organism itself. In such cases, the sequence of signal states is called a *genotype*, while the structure produced according to this genotype is called a *phenotype* of the organism, following the terminology used in biology. The smallest unit of information in a genotype is also called a *gene*. *Variation* of an organism in CA is the phenomenon where some alteration occurs to its genotypical or phenotypical structure, and it is particularly termed *mutation* if the variation is caused by a random or pseudo-random flipping of genes.

3.3 Formal definitions

Self-reproduction/replication and evolution of artificial organisms in CA has so far been defined in merely an intuitive way, while no mathematical definition has ever been given to them. However, to formulate these phenomena into some mathematical forms would be significant for clarifying description and consideration of the model. Thus, we shall make here an attempt to give them a simple mathematical definition below.

First of all, we define a *structure* as a contiguous region composed of active sites, which is represented by

$$S = \{(x_i, u_i)\} \quad (i = 1, 2, \ldots, k;\ x_i \neq x_{i'} \text{ if } i \neq i')$$

where $x_i \in Z^D$ is a coordinate of an active site while u_i ($\neq$ quiescent) $\in \Sigma$ is its state. It is furthermore required to the structure that, for all i, i' ($i \neq i'$), there must be a path from x_i to $x_{i'}$ which consists of only the edges between the adjacent sites contained in S. The last requirement guarantees this structure to be a contiguous region.

Two structures

$$A = \{(x_i^A, u_i^A)\} \quad (i = 1, 2, \ldots, k)$$

[1]As for our model introduced in the latter part of this dissertation, it can display a phylogenetic, evolutionary process caused by some variation of artificial organisms as well as an exact replication of them; therefore we intentionally use the term *self-reproduction* for representation of the behavior of our model.

and

$$B = \{(x_j^B, u_j^B)\} \quad (j = 1, 2, \ldots, k)$$

are regarded as being *congruent* with each other ($A \cong B$) if there is a vector $v \in Z^D$ which satisfies

$$\{(x_i^A + v, u_i^A)\} = \{(x_j^B, u_j^B)\}.$$

In a CA space with rotational symmetry, also $A \cong B$ if another structure A' which can be identical to A when rotated is congruent with B.

Then, we shall represent a *species* of artificial organisms in CA by using the notion of structures. Since the shape of an identical organism in CA generally changes as time proceeds, we need a set of structures which have some features common to each other in defining the species of that organism, which is represented by

$$\mathcal{S} = \{S_1, S_2, \ldots\}$$

where S_i $(i = 1, 2, \ldots)$ is a structure that has some common features.

Now, for any configuration s_t at time t, all the active sites in s_t which is represented by

$$C(s_t) = \{(x, s_t(x)) \mid s_t(x) \neq \text{quiescent}\}$$

can be divided into several structures $C_1, C_2, \ldots, C_n$ which satisfy $C_1 + C_2 + \ldots + C_n = C(s_t)$. The *population* $\mathcal{P}(s_t)$ is defined here as a set of structures $P_1, P_2, \ldots, P_m$ which are generated by the division of $C(s_t)$ that minimizes the number of structures, namely

$$\mathcal{P}(s_t) = \{P_1, P_2, \ldots, P_m\} \quad (P_1 + P_2 + \ldots + P_m = C(s_t))$$

where $m \leq n$ for all $C_1, C_2, \ldots, C_n$ which satisfy $C_1 + C_2 + \ldots + C_n = C(s_t)$. Population $\mathcal{P}(s_t)$ intuitively means the set of all structures in s_t being isolated from each other. By using $\mathcal{P}(s_t)$, the number of organisms of species $\mathcal{S}$ in s_t is defined as

$$\mathcal{N}(s_t, \mathcal{S}) = |\{P \in \mathcal{P}(s_t) \mid \exists Q \in \mathcal{S},\ Q \cong P\}|.$$

Then, finally, we can give a formal definition to the self-reproduction/replication of organisms of species $\mathcal{S}$ in CA, that is

$$\exists T > 0;\ \ \mathcal{N}(s_t, \mathcal{S}) < \mathcal{N}(s_{t+T}, \mathcal{S}).$$

Furthermore, the evolution of organisms of new species $\mathcal{S}'$ can be defined as

$$\exists T > 0;\ \ \mathcal{N}(s_t, \mathcal{S}') = 0 \ \text{ and } \ \mathcal{N}(s_{t+T}, \mathcal{S}') > 0.$$

We should note in passing that the definitions of self-reproduction/replication and evolution discussed here are both dependent on an arbitrary set of structures, i.e., species

$\mathcal{S}$. This implies that these phenomena can be observed only when the observer has some appropriate framework for interpreting the configuration of CA; therefore any kind of interpretation is possible according to the different definition of species $\mathcal{S}$. This seems to involve a significant point that what we view as life may change according to the manner of observation.

3.4 Former models of artificial life on CA

In the following several sections, the attempts so far conducted to embody artificial organisms on CA are roughly reviewed in chronological order.

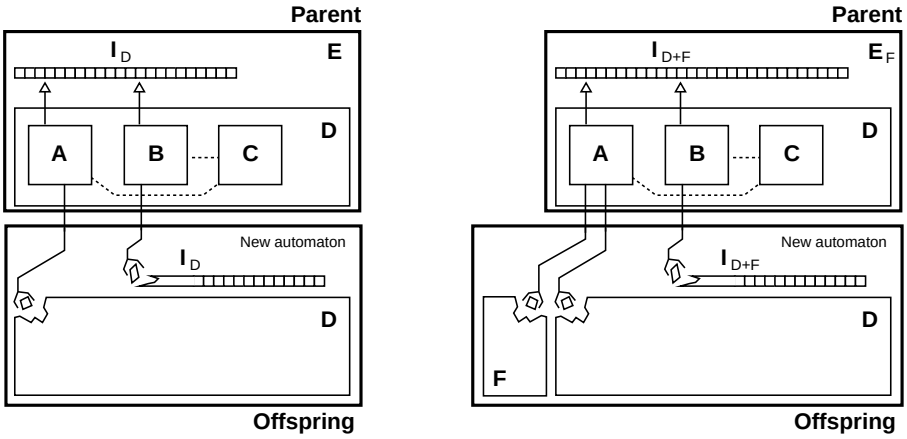
3.4.1 Creating a universal constructor-computer

The first study of creating an artificial organism on a CA space was conducted by von Neumann in 1950's[7, 58]. He devised CA, with Ulam's help, for the mathematical formulation of self-reproduction of organisms. The basic composition of his self-reproducing automaton[59] can be summarized as follows (see also Fig. 3.1):

1. Universal constructor A can construct another automaton according to instruction tape I.
2. Copier B can make a copy of instruction tape I.
3. Controller C combines A with B and functions as follows:
   (a) Let A construct another automaton according to I.
   (b) Let B make a copy of I and attach it to the automaton constructed above.
   (c) Separate the new automaton from the system $A + B + C$.
4. Automaton D consists of $A + B + C$.
5. Instruction tape I_D describes how to construct automaton D.
6. Automaton E consists of $D + I_D$, which can reproduce itself.
7. Instruction tape I_{D+F} describes how to construct automaton D and additional automaton F.
8. Automaton E_F consists of $D + I_{D+F}$, which can reproduce itself and also construct automaton F.

Von Neumann embodied this idea on a 29-state 5-neighbor CA space with weak rotational symmetry (Fig. 3.2)[58][2]. In this model, several states of the CA represented the

[2]He really could not complete this work before his death; it was posthumously completed by Burks.



A:	Universal constructor directed by instruction tape I_D
B:	Copier of instruction tape I_D
C:	Controller of A and B
D:	$A + B + C$
I_D:	Instruction tape which directs how to construct D
E:	$D + I_D$
I_{D+F}:	Instruction tape which directs how to construct $D + F$
E_F:	$D + I_{D+F}$
F:	Additional automaton

Figure 3.1: Schematic illustrations of the idea of von Neumann's self-reproducing automaton model[59]. This consists of three automaton (universal constructor A, tape copier B, and controller C; $A + B + C = D$) and instruction tape I_D. Since I_D contains the information about how to construct the automaton D itself, the whole system $E = D + I_D$ can reproduce itself (left). In addition, if the instruction tape contains additional information about another automaton F, then the whole system $E_F = D + I_{D+F}$ produces F as well as itself (right), which can be considered as a kind of evolution.

wires of virtual logical circuits, while others represented the signals conducted in them. The state-transition rules were carefully designed so as to realize their operation. This automaton had the capability of universal construction/computation—the ability to construct any arbitrary structure and execute any computational operation directed by the given instruction tape—and if set up with an instruction tape describing how to construct the automaton itself, it could reproduce itself. It has been regarded as the first theoretical model of self-reproduction. Why this self-reproductive model was so important is that it predicted the self-reproductive mechanism of real life composed of genetic information (DNA) and its translators/transcribers (cytoplasm) before the discovery of double helix structure of DNA in 1953.

After von Neumann's work, several variations were invented. For example, Arbib presented another cellular space composed of relatively complex programmable cells where implementing self-reproducing automata was easier than in von Neumann's CA[3]. Codd simplified von Neumann's universal automaton onto much simpler 8-state 5-neighbor CA with strong rotational symmetry[11]. Vitányi created sexually reproducing automata in the same 8-state 5-neighbor CA with strong rotational symmetry as Codd's (Fig. 3.3)[57]. However, these models including von Neumann's were all in prohibitive size, due mainly to their constructional universality, so it was impossible at that time to simulate them on computer[3]. They merely remained theoretical works.

3.4.2 Searching for a minimal system capable of non-trivial self-replication

The next epoch-making work was done by Langton in 1984[27]. He thought of the constructional universality of automata as sufficient but not necessary condition for self-reproduction, and claimed that if the information structure explicitly directing self-reproduction existed in the structure of the organism, and if it was used in two different ways—translation and transcription, then we could regard the self-reproductive process as non-trivial[4] one. According to this new criterion, he proposed a self-reproducing loop in extremely small size (Fig. 3.4) compared to the above-mentioned universal automata, by modifying a periodic emitter which had been a component of Codd's automaton[11]. This model has sometimes been called a "sheathed" loop since its structure was covered by the "sheath" states. The details of this model are introduced in section 4.4.

Langton's claim was eagerly taken up by many researchers, and the minimal CA system capable of non-trivial self-reproduction was speculated on through the simplification of

[3]Von Neumann's automaton was realized later in computer simulation by Pesavento[39], as shown in Figure 3.2.

[4]Its counterpart is trivial self-replication, represented by the example of the XOR rule introduced in section 2.3.

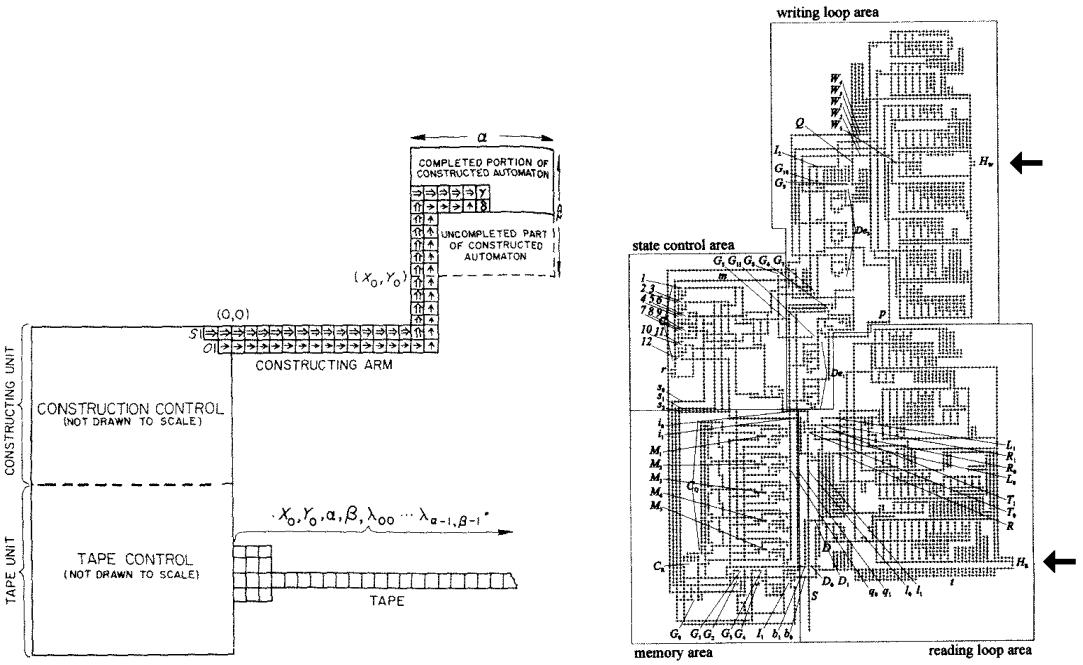


Figure 3.2: Von Neumann's self-reproducing automaton implemented on a 29-state 5-neighbor CA space. Left: The illustration of von Neumann's idea (from [7]). The box indicated as "CONSTRUCTION CONTROL/TAPE CONTROL" corresponds to automaton D, and the "TAPE" drawn at the lower-right part of the figure corresponds to instruction tape I_D, in Figure 3.1. This automaton reproduces another automaton (the upper-right part of the figure) according to the direction contained in the tape, and if set up with the tape describing how to construct the automaton itself, it can reproduce itself. Right: Its detailed structure realized by Pesavento (from [39]). Each dot in this figure indicates a state on a single site of the CA. The instruction tape is attached to an input port of the reading loop area (pointed by the lower arrow). The information written in the tape is interpreted by the logical circuits implemented in the body of the automaton, and finally the result of interpretation is sent outward through the construction arm which is projected from an output port of the writing loop area (pointed by the upper arrow).

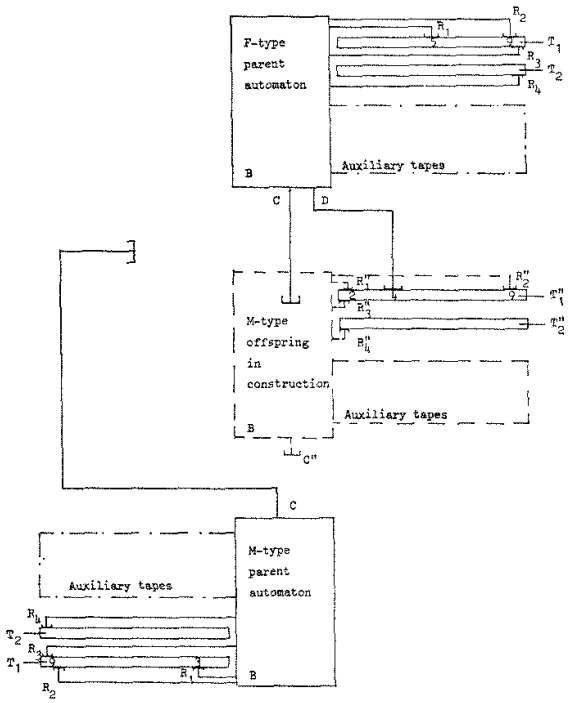


Figure 3.3: Vitányi's sexually reproducing automata (from [57]). These automata have construction arms and instruction tapes like von Neumann's automaton, but their characteristic point is that the offspring automaton (middle) is constructed by the cooperation of two different kinds of the parent automata, i.e., F-type (upper) and M-type (lower).

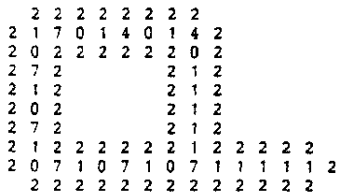


Figure 3.4: Langton's self-reproducing loop (from [27]). This model was implemented on 8-state CA with a von Neumann neighborhood. A signal sequence of '7-0 7-0 7-0 7-0 7-0 7-0 4-0 4-0' is set in a 'Q'-shaped tube structure enclosed by sheath state '2's. Its details are introduced in section 4.4.

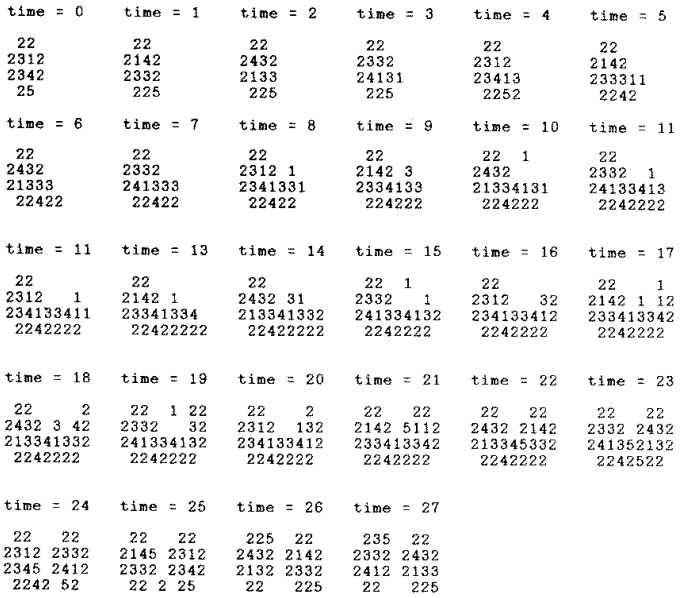


Figure 3.5: Self-reproductive process of Byl's small loop (from [8]). This model was implemented on 6-state CA with a von Neumann neighborhood. The initial structure of this loop consists of only 12 sites in active states, where a signal sequence of '3-3-1-4' is folded and enclosed by sheath state '2's. This sequence circulates counterclockwise in the sheath structure, while its copy is sent outward and translated into the structure of an offspring loop. After 25 updates, a new offspring whose shape is identical to the initial structure is produced in the right area of the parent. This manner of self-reproduction followed that of Langton's loop introduced in section 4.4.

Langton's loop. For example, Byl invented a much smaller loop than Langton's which consisted of 12 sites in a 6-state CA space (Fig. 3.5)[8]. Reggia et al. considered a variety of self-replicating loops with various properties, e.g., sheathed as well as unsheathed, with strong rotational symmetry as well as weak, with a von Neumann neighborhood as well as Moore, and so on (Fig. 3.6)[42].

Besides the modification of Langton's loop, there were several interesting challenges to realize self-reproduction in other forms. Sipper showed that it is much easier to implement many lifelike phenomena by using non-uniform non-deterministic CA than by using traditional CA[49]. Morita et al. succeeded in creating self-reproducing structures with a shape-encoding mechanism with which every structure (worms as well as loops) can reproduce itself in a non-trivial way (Fig. 3.7)[32, 33].

3.4.3 Applying to practical problems

The application of self-replicating automata models for the solution of practical problems has been energetically studied too. Tempesti invented a semi-sheathed loop (which kept

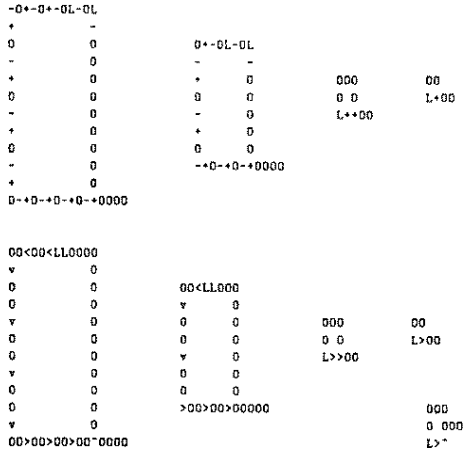


Figure 3.6: Various kinds of unsheathed self-replicating loops discovered by Reggia et al. (from [42]) The spaces used here are all 8-state CA with a von Neumann neighborhood. The upper four loops are implemented on CA with strong rotational symmetry, and the lower five are on CA with weak rotational symmetry. Each loop shown here originally forms a perfect square, which is distorted in this figure by a tall character font.

sheath states only on the inner side of the loop) after Langton's model, which could contain an additional program in the signal stream on the surface of its body, and also could construct some additional structures within itself according to the given program, while reproducing itself[53] (Fig. 3.8).

Perrier et al. succeeded in furnishing self-reproducing loops with the computational universality[38]. They attached a static program tape and an elastic data tape on the surface of Langton's loop (Fig. 3.9), which functions as a computationally-universal W-machine[5]. They demonstrated that this loop could actually execute the program to check the matching of parentheses contained in a given string, as well as the correct self-reproduction of the whole system.

Chou et al. tried to apply self-replicating loops for solving a satisfiability problem[10]. This problem is to judge whether there is an assignment of values to the logical variables which satisfies a given logical formula (i.e., which makes a truth-value of that formula be 'True'). This problem generally involves the NP-complete complexity; namely, the order of time necessary for solving this problem is believed to be greater than a polynomial time in general. Chou et al. aimed at solving a problem of an exponential ($\mathcal{O}(2^n)$) size through the effective use of 2-dimensional ($\mathcal{O}(n^2)$) space. They overlapped an ordinary CA space

[5]The W-machine[60] is a variant of the Turing machine[24], proposed by Wang. Its computational capability is completely the same as that of the Turing machine, while its style of computation directed by a certain instruction set is more suitable for practical programming than the state-transition diagram used in the Turing machine.

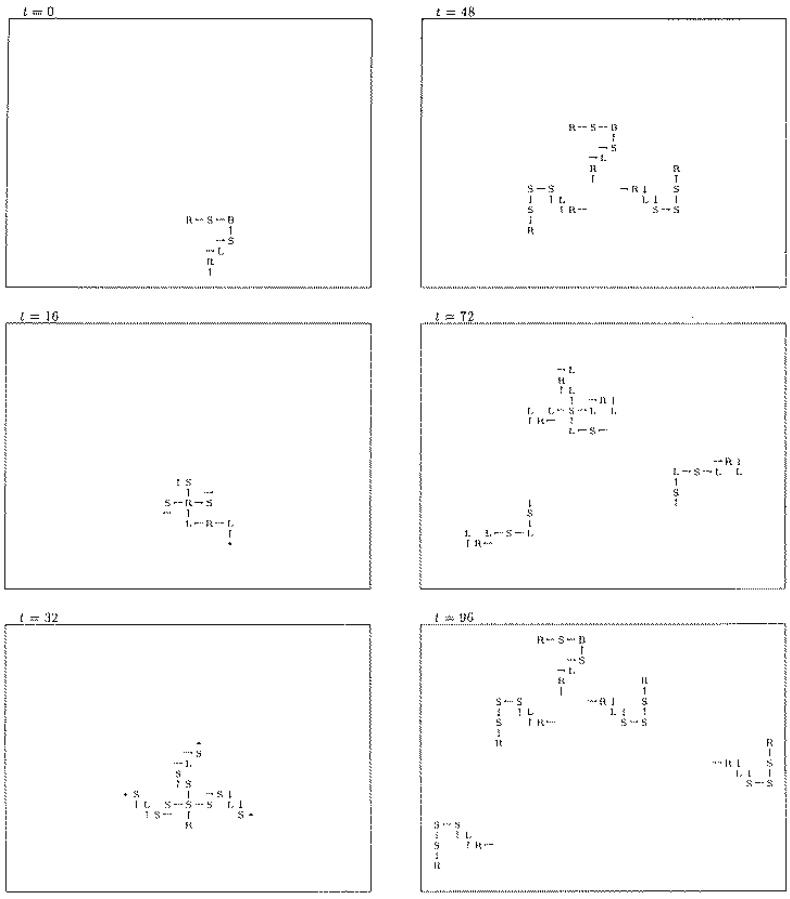


Figure 3.7: Morita et al.'s self-reproducing worm with a shape-encoding mechanism (from [33]). With this mechanism, the organisms are always encoding their own shape into their genotype, thus every structure (worms as well as loops) can reproduce itself in a non-trivial way in this world. This figure shows a worm which continuously encodes its shape into its genotype at the tail of its own. These encoded signals are conveyed from the tail to the head, where they are utilized for construction of the structures of its offsprings.

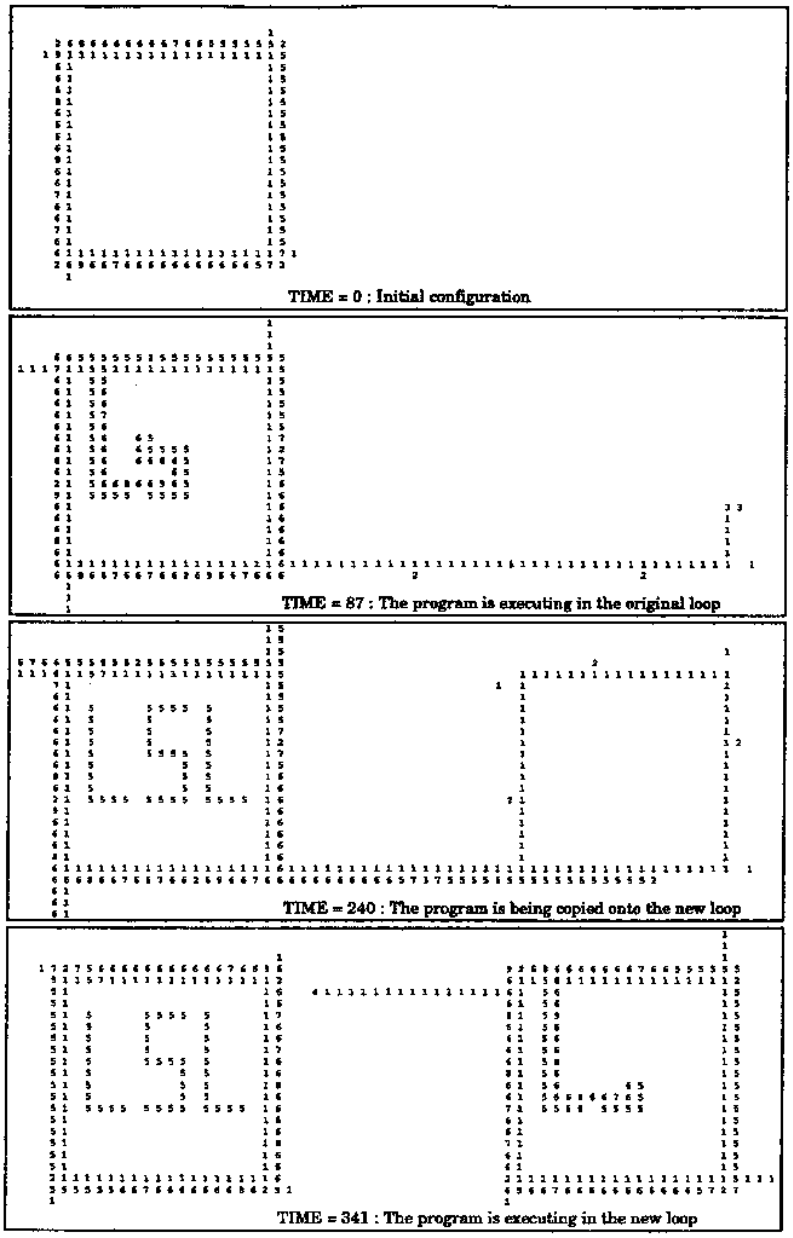


Figure 3.8: Tempesti's self-reproducing loop capable of additional construction (from [53]). This loop can contain an additional program in the signal stream on the surface of its body and can construct some additional structures within itself according to the given program, while reproducing itself. In this case, the loop constructs the letters 'LSL' in its body, and this feature is inherited by its offspring.

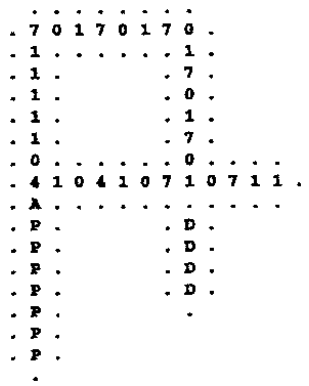


Figure 3.9: Perrier et al.'s self-reproducing loop capable of universal computation (from [38]). A static program tape (indicated by 'P') and an elastic data tape (indicated by 'D') were attached to the surface of the loop which was identical to Langton's. These devices functioned as a computationally-universal W-machine after self-reproduction of the loop.

upon an array of "monitor" automata. When an ancestral loop initially containing unexplored logical variables begins to self-replicate in this space, these variables are gradually explored to a concrete assignment through self-replication. The loop having an unsatisfiable assignment becomes eliminated by the monitor automata, and finally, only the loops having the satisfiable assignments can survive in the space (Fig. 3.10), which mean the solutions of the problem. This work presented a new perspective about the application of self-replicating structures, which may resemble the DNA computing[2].

The hardware implementation of self-replicating structures are also in progress by several groups, e.g., Sipper et al.[51]

3.4.4 Realizing emergence and evolution

The most recent issue in this research field may be to realize the emergence and evolution of self-replicating structures in the CA space.

Lohn et al. developed an automatic discovery system of state-transition rules of CA where a given structure could replicate itself in the space (Fig. 3.11)[30]. In this work, an initial structure was *a priori* given while the state-transition rules themselves evolved to make the structure self-replicate, through the genetic algorithms (GA)[19, 23] customized for this task.

Chou et al. simulated an emergent process of self-replicating loops from the "primordial soup" of the states of CA[9]. They used 256-state multi-data-field CA (in which the information included in a state value was divided into some data fields and treated separately) with a Moore neighborhood, where states were separated into two categories: Bound and unbound. The bound states yielded to the normal state-transition rules which

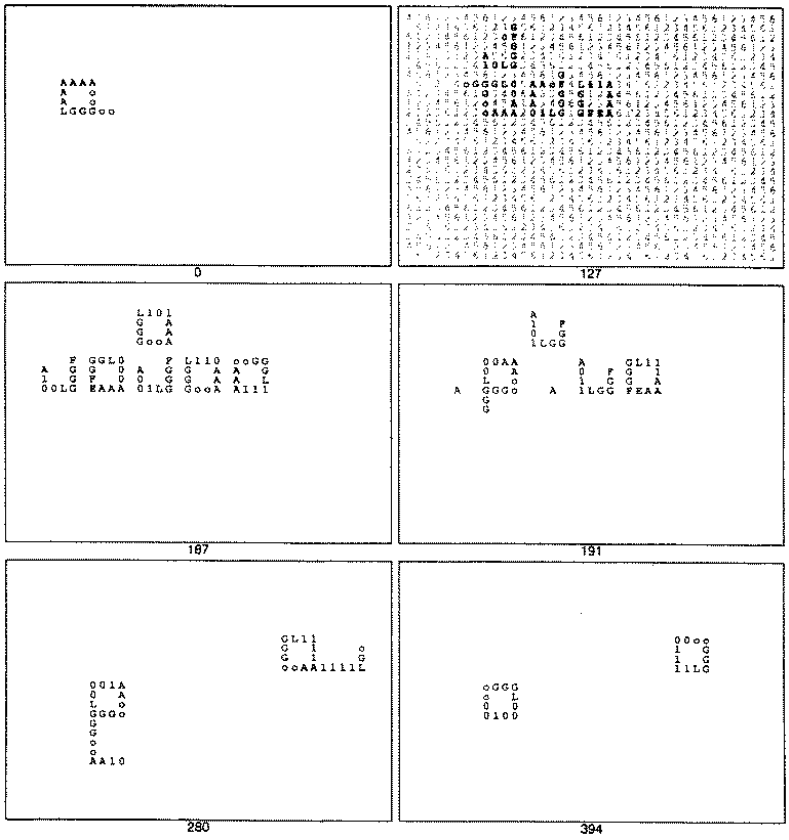


Figure 3.10: Chou et al.'s self-replicating loops solving a simple satisfiability problem (from [10]). The monitor automata are drawn in only a upper-right figure. The simulation begins with an ancestral loop initially containing unexplored logical variables (indicated by 'A'). These variables are gradually explored to a concrete assignment ('0' or '1') through self-replication. The loop having an unsatisfiable assignment becomes eliminated by the monitor automata, and finally, only the loops having the satisfiable assignments survive, which mean the solutions of this problem.

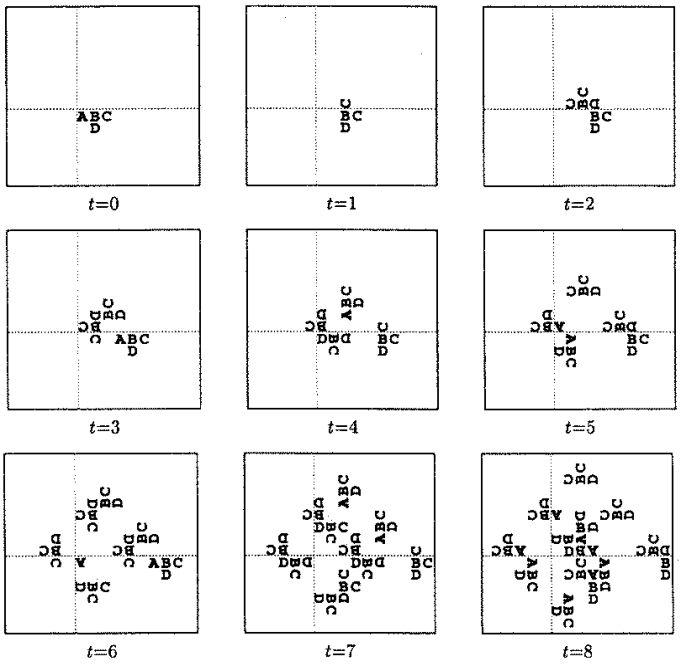


Figure 3.11: Example of Lohn et al.'s self-replicating structures governed by the state-transition rules automatically discovered by genetic algorithms (GA) (from [30]). The initial structure which is composed of four active sites rotates counterclockwise and produces its replicas to the outer area.

controlled self-replicative behavior of the loops, while the unbound ones yielded to the other Game-of-Life-like rules which realized perturbation of the primordial soup. In this space, if a structure of the minimal self-replicating loop consisting of four sites appeared in unbound states by accident, then it automatically switched from unbound to bound. Once such a small loop appeared, it gradually evolved and grew in size through the repetition of self-replication (Fig. 3.12), since Chou et al. forced loops to mutate to a larger size by their ingenious state-transition rules. Thus, this evolutionary path began at the optimum (i.e., smallest structure) of the fitness landscape and descended from there, so their evolutionary system remained somewhat unnatural. However, this work was important in being the first to realize the emergent process of self-replicating structures, and it is also notable that the dissolution of self-replicators was realized for the first time in this work.

The studies reviewed above are only a part of various artificial life models constructed on CA, of course, and many other ones have been proposed so far too. However, truly spontaneous evolution where self-replicators vary and evolve toward fitter ones through natural selection has never been attained, which is the very target of our study being discussed in this dissertation.

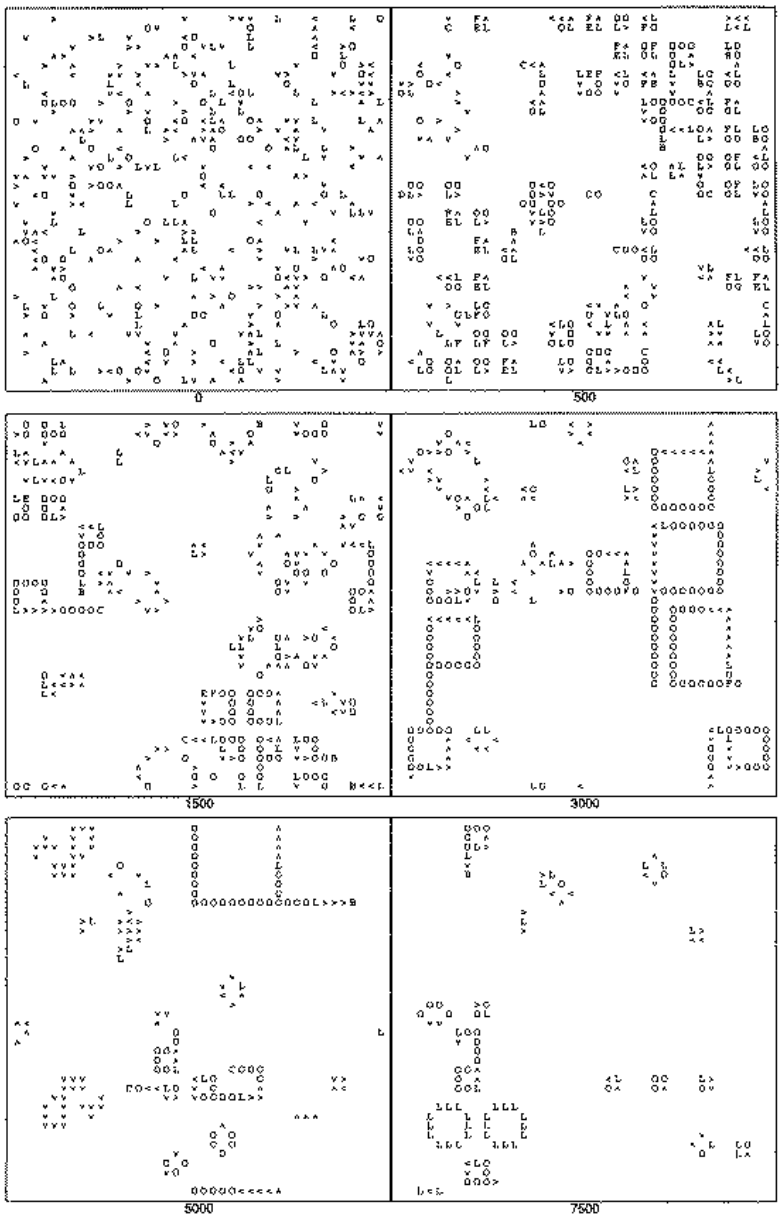


Figure 3.12: Chou et al.'s CA space where self-replicating loops emerge and evolve (from [9]). Initially, the whole space is filled with a "primordial soup", i.e., unbound states being perturbed by the Game-of-Life-like rules. When a structure of the minimal self-replicating loop consisting of four sites appears in the soup by accident, it automatically switches from unbound to bound and begins to obey the other state-transition rules which control self-replicative behavior of the loops. Once such a small loop appears, it gradually evolves and grows in size through the repetition of self-replication.

Chapter 4

STRATEGY OF THIS STUDY

In this chapter, we describe the strategy we employed in this study for the purpose of constructing evolutionary systems on a simple deterministic CA space. We also give a detailed account of Langton's self-reproducing loop which we adopted as a subject matter of this study.

4.1 Target

It would be worth doing to clarify the target of this study at first. As mentioned at the end of the previous chapter, our target is to realize the spontaneous evolution of artificial organisms on a CA space. The word "evolution" used here means a temporally developing phenomenon performed by a number of self-replicators, where the forms or functions of self-replicators as well as the composition of the whole colony develop through variation of self-replicators and natural selection acting on them.

Here, we introduce the famous artificial life *Tierra*[41] presented by Ray in 1990 as a successful example of artificial evolutionary systems formerly proposed. It was a kind of virtual operating system imitating the MIMD (multi-instruction-multi-data computer) environment on a single PC, where digital creatures (actually programs written in a virtual language) were populated in an identical finite memory space and executed in parallel. Some computational resources such as CPU time and registers were assigned to each creature by the operating system. Death and probabilistic mutation were also provided to each creature. In this system, when a simulation began with a sole ancestral creature in the memory which had only the ability to reproduce itself, then its descendants spontaneously evolved through mutation and natural selection and generated a very diverse, ecological society. Figure 4.1 shows a metabolic flowchart of some creatures interacting with each other, which evolved to emerge in the world of *Tierra*.

Our final goal is, so to speak, to realize such an evolutionary system like *Tierra* in a CA space. If we could succeed in such transplantation, the whole system would become very

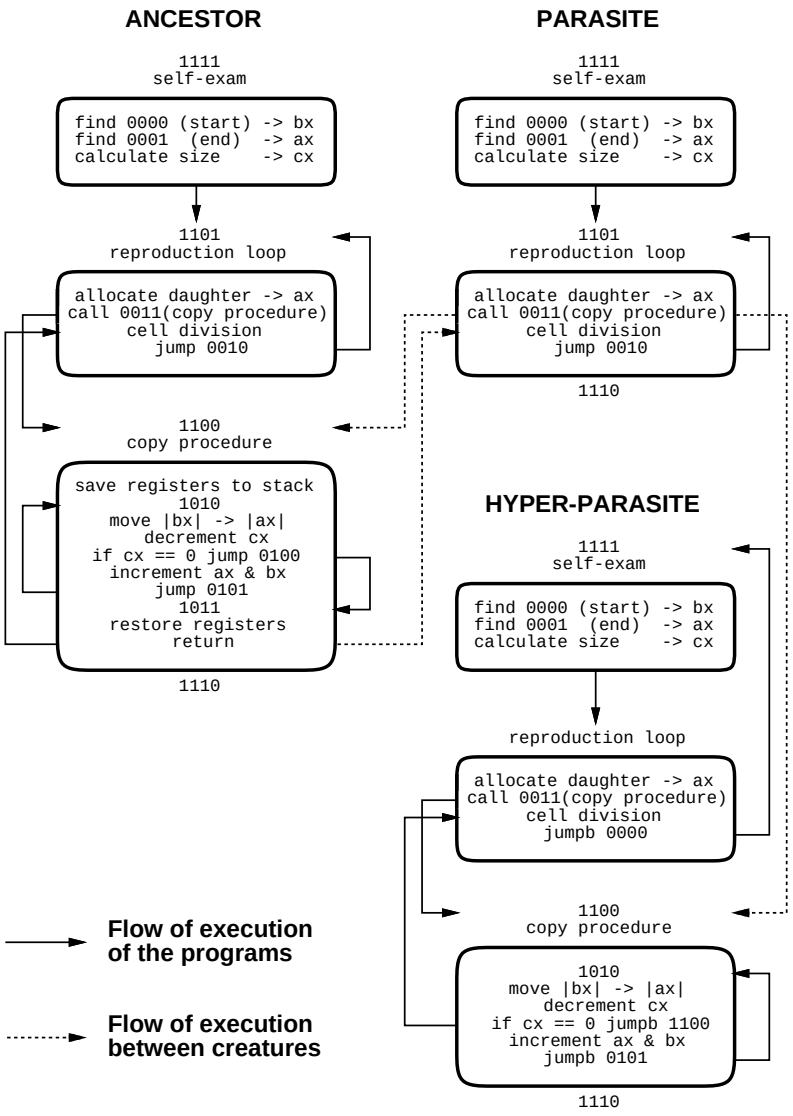


Figure 4.1: Metabolic flowchart of creatures that evolved in the world of *Tierra* (from [41]). Each creature is really a computer program located in an identical memory space, which is generally comprised of some modules indicated by rounded boxes in this figure. Arrows between the boxes indicate the flow of execution of the programs. The ancestral program (left) consists of three modules: A preprocessing module which measures the size of the whole program, a main loop which repeats allocating of an enough memory space for self-reproduction and copying of the instructions in itself into the allocated memory, and a copy procedure called by the main loop. When a simulation begins with a single ancestral program, various kinds of variants evolve and emerge, such as a parasite which exploits the copy procedure of the ancestral species (upper right), and even a hyper-parasite which catches the parasite's instruction pointer (which is considered as the energy for the programs in this world) to utilize for the reproduction of itself (lower right). Many other variants were also observed in the world of *Tierra*.

simple and suitable for superparallel processing; then, it would also become possible to enlarge the scale of simulation enormously in the future by implementing such a system on minute hardware devices, from which we could obtain further knowledge on the evolution of *life-as-it-could-be*.

4.2 Problems and basic ideas for solution

The evolution of life generally consists of two essential events occurring in the population, i.e., variation and natural selection, which have been missing so far in most CA models. For the implementation of evolution on CA, it is necessary to embody these events in a CA space somehow or other.

Variation seems comparatively easier to introduce into CA than natural selection, because the state transition occurring in only one site which contains genotypical information may cause the variation of the organism[1]. Such state transition in a single site is tractable in CA.

On the other hand, natural selection inevitably involves the notions of fitness and death of organisms. As surveyed in the previous chapter, there was no model of artificial life on CA equipped with both of these notions. Thus, we decided to try to introduce natural selection into the CA first. In this attempt, the state-transition rules from active to quiescent states call for a lot of devices, being opposed to former studies in which only the rules from quiescent to active states have been focused on (Fig. 4.2), because the death of organisms in CA must be realized by the state-transition rules from active to quiescent states.

We should note here that the transition of a state into a quiescent one occurring in a single site is not regarded as death of life in CA, because even a sole individual of artificial life on CA is generally represented by a contiguous structure which is comprised of many active sites in the space. Death of an organism in CA should be the phenomenon where all active sites in such a structure turn into a quiescent state by some mechanism.

In order to realize the elimination of a contiguous structure from the space, it is necessary to generate the information indicating the death of that structure and transmit it to all sites in that structure. In devising this mechanism, the idea of a "decomposing" automaton in Oohashi et al.'s programmed self-decomposition model[35, 36] would serve as a good reference. This model is an ecological hypothesis which assumes that every cell of terrestrial creatures has intrinsically installed with the fundamental ability to decompose itself and return the space and substance which it had possessed to the environment. In this model, a biological cell is represented as a variation of von Neumann's self-reproducing automaton, as shown in Figure 4.3, which contains an additional "decomposing" automa-

[1]But such a flipping of genes will prove not necessary for the realization of variation in chapter 6.

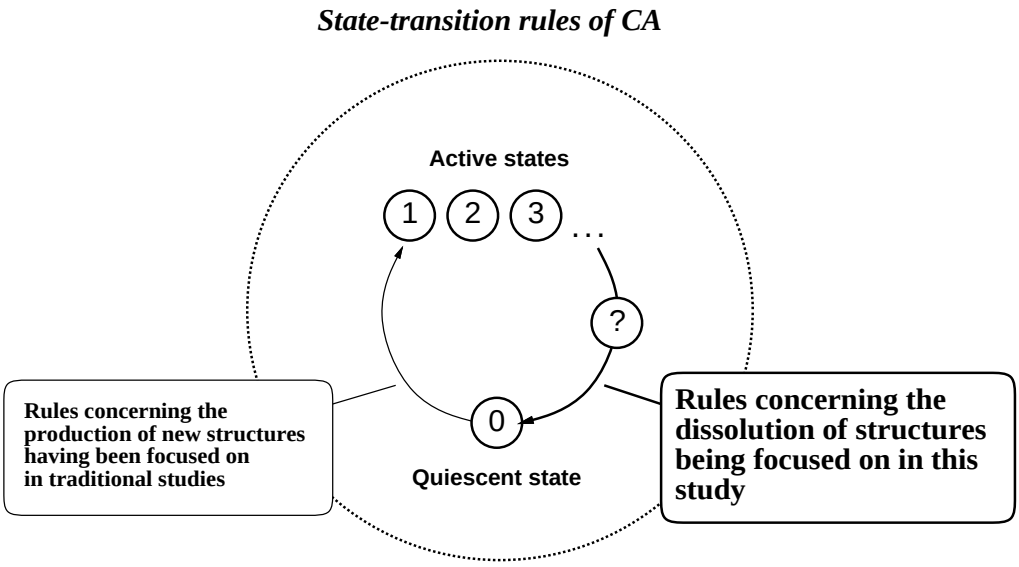
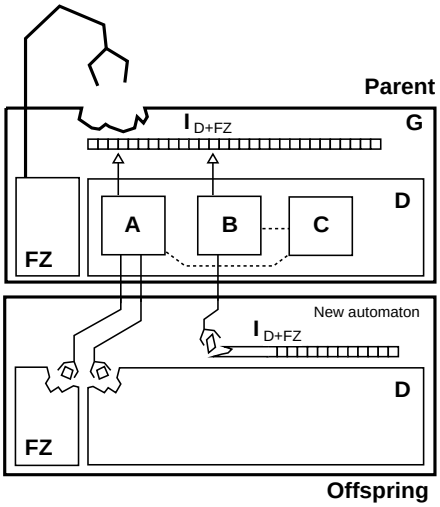


Figure 4.2: Schematic illustration of the difference between this study and other former ones with respect to which part of state-transition rules are mainly focused on. Traditional studies of artificial life on CA has focused on the rules from quiescent to active states which are mainly concerned with self-reproduction (left), while this study focuses on the rules from active to quiescent states which are necessary for the realization of death and natural selection of organisms in a CA space (right).



A:	Universal constructor directed by instruction tape I_{D+FZ}
B:	Copier of instruction tape I_{D+FZ}
C:	Controller of A and B
D:	$A + B + C$
I_{D+F}:	Instruction tape which directs how to construct $D + FZ$
FZ:	Decomposing automaton
G:	$D + FZ + I_{D+FZ}$

Figure 4.3: Oohashi et al.'s self-reproducing/decomposing automaton model[35]. Compared with von Neumann's (Fig. 3.1), this contains an additional "decomposing" automaton FZ which has the ability to decompose the whole system to its constituent elements. FZ autonomously awakes to function when it finds the whole system inconsistent with the external environment.

ton FZ as well as the same components as von Neumann's model has. FZ has the ability to decompose the whole system completely to its constituent elements. It autonomously awakes to function when it finds the whole system inconsistent with the external environment. The author formerly has experienced the development of artificial life simulators based on this hypothesis[36, 43], where the decomposing automaton worked effectively for detection and elimination of individuals inconsistent with the environment. The idea of decomposing automaton FZ which autonomously emerges in the body of such an unfit individual and decomposes its structure is much helpful in designing how to realize the death of organisms in CA.

Referring to the above idea, we planned to realize the death of organisms in CA by a new *dissolving state* emerging and sweeping over the structure of the organisms. To be more specific, we devised the dissolution of a contiguous structure on a CA space through the emergence and propagation of a new state which will turn into a quiescent state after one update. The method of introducing the new state into CA is planned as follows:

1. Introduce a dissolving state '**d**' into the set of states of the CA.

2. Let '**d**' turn into a quiescent state at any time.

3. Define rules which realize the contagion of '**d**'.

4. Define rules which realize the dissolution of the structures next to '**d**'.

5. Let all the states in undefined situations turn into '**d**'.

1, 2 and 5 of this procedure are directly applicable to any kind of CA, while 3 and 4 need some model-dependent invention. The point of this method is 5, which aims at embedding the fitness function into the state-transition rules; in other words, the fitness of an organism in this study is whether it can avoid facing undefined situations. When the organism unfortunately faces such a situation, a dissolving state '**d**' emerges to make the organism die. Thus, if this method would be appropriately applied, the CA would be capable of quick elimination of unwanted structures from the space by generation and propagation of the dissolving state '**d**'.

4.3 Distinction between two aspects of death

We would like to point out here, related to introduction of death into CA, that two distinct aspects are involved in what is called death.

To give an answer to the question what kind of behavior we call death, at least we can say as a trivial definition that death is a state transition of a system from a living state to a non-living one. Though we also need definitions for the living state and the non-living state in order to complete this statement, we do not want to consider it too deeply here. For the time being, here we define the living state to be a state in which a system functions correctly and shows complex, adaptive behavior. Death of life as a state transition from the living state to the non-living caused by loss of sound function of the system is referred to as *death as functional failure* in this dissertation.

In practice, however, there are many cases in which we view the cessation of life not only as functional failure but also by the disorganization of physical structures of a system. For example, all terrestrial creatures including human beings decompose into organic compounds by various chemical reactions and other microorganisms after death due to functional failure, unless a special treatment for preservation is applied to their corpses. Moreover, it is clearly understood that in the engineering techniques such as genetic algorithms (GA)[19, 23] and genetic programming (GP)[26], or in artificial ecosystem simulations like the above-mentioned *Tierra*, death of an individual implies not only their functional termination but also removal of them from the memory space of computers.

Thus, it is obvious that the value of death for a system in a finite environment would not only be functional failure, but in fact in the disorganization of structures associated

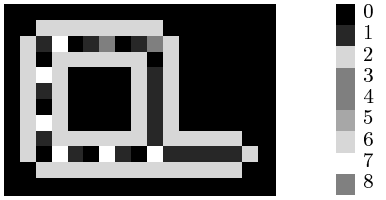


Figure 4.4: Initial structure of Langton's SR loop. Six gene '7's (each of which makes the arm grow straight for one site) and two gene '4's (which make the arm turn left) are set in a 'Q'-shaped tube structure enclosed by sheath '2's. The right chart indicates the correspondence between states of sites and shades of pixels in the figure. The following figures are also drawn according to this chart.

with it. The latter is the very death we are going to introduce into the artificial organisms on CA, which is referred to as *death as structural dissolution* in this dissertation.

4.4 Subject matter: Langton's self-reproducing loop

For the objective of this dissertation to construct evolving artificial organisms on a CA space, it is considered that to make good use of some existing CA model, in which the fundamental behavior of self-reproduction has already been realized, is far more efficient than to work out another novel model from scratch. In accordance with this idea, we chose Langton's self-reproducing (SR) loop[27] as a subject matter of this study. The reason of this choice is that

1. Langton's loop is very famous and its behavior has been studied well,
2. its CA space is quite simple (an 8-state CA space with a von Neumann neighborhood) which is a favorable feature for future hardware implementation, and
3. it resembles some aspects of real organisms such as having both genotype and phenotype separated from each other in the CA space.

Langton's SR loop is one of the most famous models of self-reproduction constructed by means of CA. It is implemented in a simple 8-state CA space with a von Neumann neighborhood and strong rotational symmetry. Figure 4.4 shows its initial structure, where gene '4's and '7's are set to float in a 'Q'-shaped tube enclosed by sheath '2's. Names and functions of the states in this CA are listed in Table 4.1.

Each gene travels along the tube in the direction of the neighboring core '1', and splits into two identical genes at the T-junction of the tube structure. One of them circulates into the loop again and the other goes down toward the tip of a construction arm which is thrust outward from the loop. When a gene reaches the tip of the arm, translation

Table 4.1: Names (temporarily used in this dissertation) and functions of the states in the CA of Langton's SR loop.

Fundamental elements

State	Name	Functions
'0'	Background	The quiescent state of this CA.
'1'	Core	To fill the tube of the loop and conduct signals in it.
'2'	Sheath	To form the tube structure of the loop.

Signals

State	Name	Functions
'3'	Left indicator	To support left turning of the arm.
	Bonder	To support bonding of the two arms.
	Sprout generator	To support germinating of the offspring's new sprout.
	Sprout capper	To cap the tip of the offspring's sprout.
'4'	Gene	To keep genetic information of left turning of the arm and finishing of growth of the offspring's sprout.
'5'	Umbilical cord dissolver	To dissolve the umbilical cord between parent and offspring.
	Messenger	To point where the parent should germinate a new sprout.
	Sprout guide	To support growth of the parent's sprout.
'6'	Messenger	To point where the offspring should germinate a new sprout.
	Sprout guide	To support growth of the offspring's sprout.
	Sprout finisher	To finish growth of the offspring's sprout.
'7'	Gene	To keep genetic information of straight growth of the arm and the offspring's sprout.

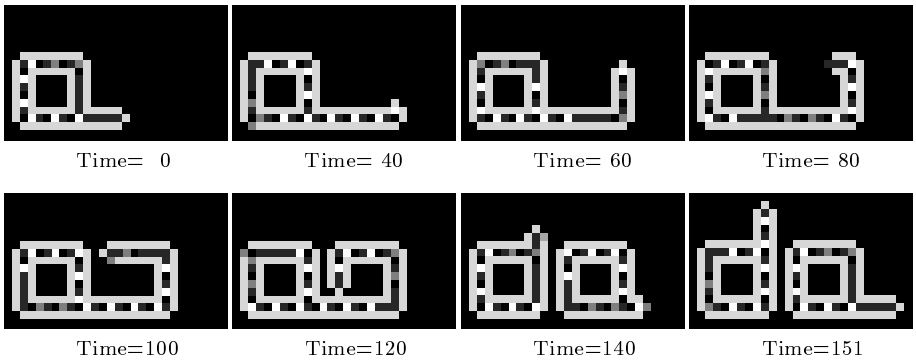


Figure 4.5: Self-reproduction of the SR loop. As genes circulate in the loop counterclockwise, copies of them are made and sent toward the tip of the arm. The arm grows through repetition of straight growth and left turning (time=40–100). When the tip reaches its own root after three left turns, they bond together to form a new offspring loop. Then the connection between parent and offspring disappears (time=120–140). In such a way, the loop reproduces its offspring which has a structure identical to its parent's in the right area, in 151 updates (time=151).

from genotype to phenotype will occur, such as straight growth or left turning of the arm. When the tip of the arm reaches its own root after it turned left three times, the tip and the root bond together to form a new offspring loop, and then the connection between parent and offspring—which Langton called the "umbilical cord"—disappears. The SR loop is ingeniously designed to reproduce itself in just 151 updates. This method of self-reproduction is shown in Figure 4.5. The detailed mechanisms of several scenes in its self-reproductive process are also depicted in Figure 4.6–4.10. The complete set of the state-transition rules of Langton's loop is shown in appendix A.

When the SR loop finishes self-reproduction, it will try to do the same again in the same way but rotated by ninety degrees counterclockwise. It repeats this action indefinitely while there is enough space for self-reproduction. If an area in which the loop wants to place its offspring is already occupied by others, it generates a new sheath fragment in the tube to obstruct the pathway of genes. Then, genes in the loop become extinct one after another, being absorbed by the obstacle fragment. Finally, only a circular tube filled with cores is left by itself. The manner by which this proceeds is shown in Figure 4.11. The detailed mechanisms of the beginning of this dying process is also depicted in Figure 4.12. Thus, death as functional failure is certainly present in the SR loop. However, it is not equipped with the capability of death as structural dissolution, except for partial dissolution: The disappearance of the umbilical cord.

Figure 4.13 depicts the manner by which the colony of the SR loop grows. It shows a coral-like growth where its inner area becomes filled with dead structures while the loops on the surface of the colony are alive.

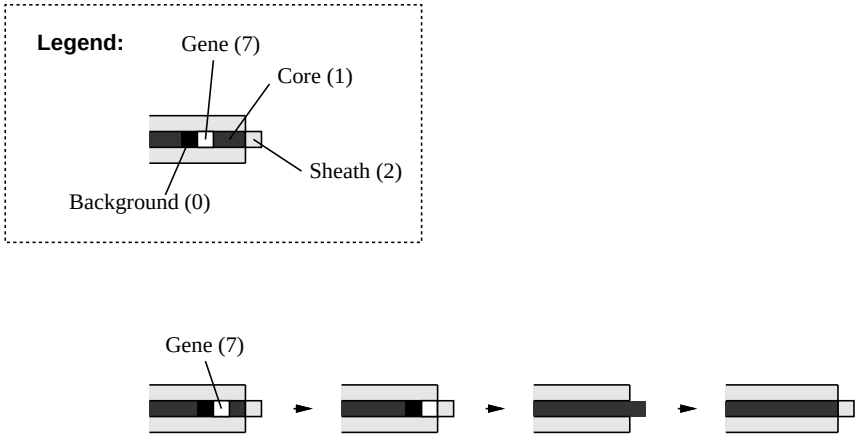


Figure 4.6: Mechanism of straight growth of the construction arm of the SR loop. Sites not explicitly specified are in background '0'. When a gene '7' traveling in the tube structure hits the tip of the arm, a sheath '2' having capped the tip turns into a core '1', and the gene '7' disappears. Then, the unwrapped core '1' quickly becomes sheathed after one update; thus, the arm becomes one site longer than before by a gene '7'.

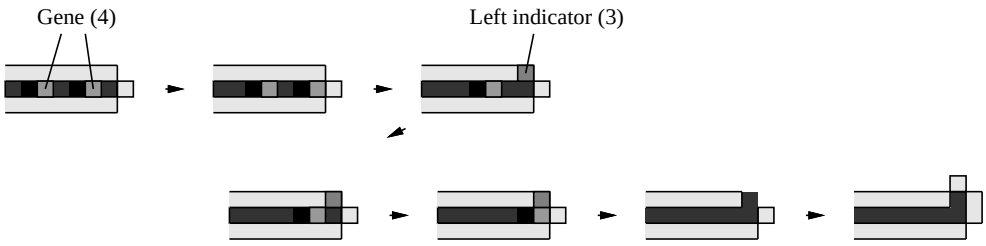


Figure 4.7: Mechanism of left turning of the tip of the arm of the SR loop. When the first gene '4' traveling in the tube structure hits the tip of the arm, a left indicator '3' emerges on the left-hand sheath of the tip, and the gene '4' disappears. This left indicator '3' remains as is until it turns into a core '1' when it is hit by the second gene '4'. The second '4' also disappears after hitting the tip. Then, the unwrapped core '1' becomes sheathed after one update. Consequently, a corner of tube structure is formed by two gene '4's.

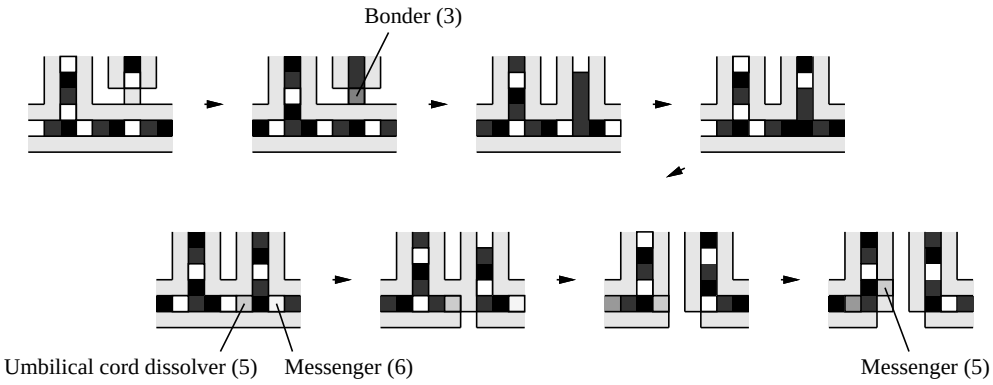


Figure 4.8: Mechanism of bonding of the tip and the root of the arm followed by the dissolution of the umbilical cord in the SR loop. When a gene '7' hits the tip touching the middle of the arm, the sheath '2' having capped the tip turns into a bonder '3', while the gene '7' disappears. After one update, the tip and the middle of the arm bond together to form a new T-junction. Then, another gene '7' which comes into the new T-junction alters into background '0', thus two-site blank space emerges at the T-junction, which generates next an umbilical cord dissolver '5' and a messenger '6'. The messenger '6' begins to travel in the tube toward the next corner, while the umbilical cord dissolver '5' goes back and dissolves the umbilical cord, and eventually it turns into a messenger '5'.

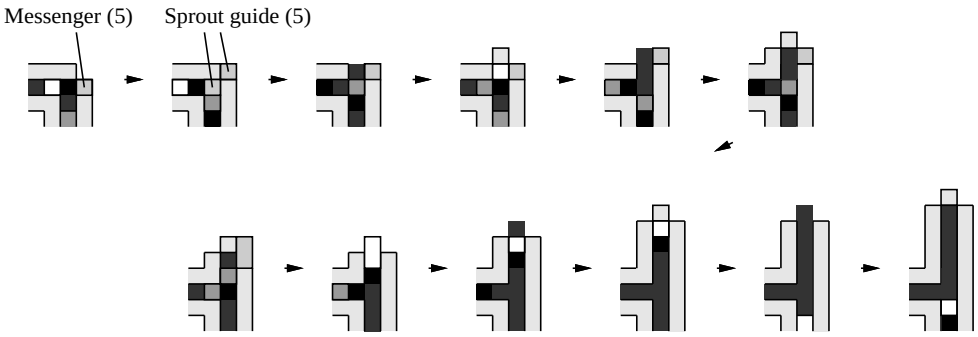


Figure 4.9: Mechanism of germination of a new sprout of the arm in the parent SR loop. When the messenger '5' having been generated in the above-mentioned mechanism (Fig. 4.8) arrives at the next corner, it turns into two sprout guide '5's. One of them quickly disappears, while the other remains on the sheath and guides a new sprout to germinate. The manner of this germination is constructed in such a heuristic way completely dependent on some specific situations that it is difficult to give any simple explanation for it.

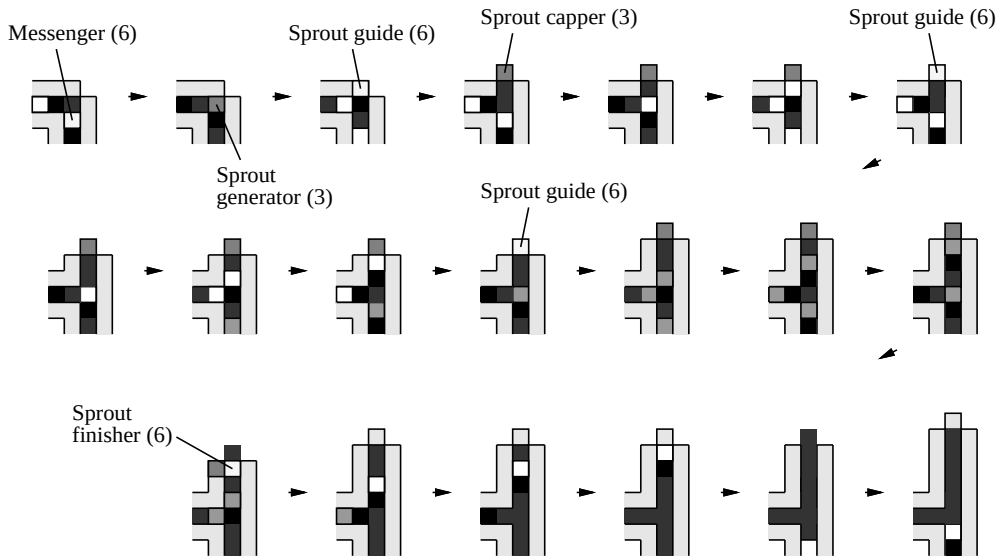


Figure 4.10: Mechanism of germination of a new sprout of the arm in the offspring SR loop. When the messenger '6' having been generated in the above-mentioned mechanism (Fig. 4.8) arrives at the next corner, it alters into a sprout generator '3'. Then, next, the gene '7' recovers from the sprout generator '3', a sprout guide '6' appears onto the outside sheath out of the sprout generator '3' , and the sprout generator itself disappears. The sprout guide '6' quickly turns into a new sprout of the arm capped by a sprout capper '3'. This sprout is stimulated to grow two times by gene '7's. When a gene '4' arrives at the tip of the sprout, it alters into a sprout finisher '6', which changes the tip of the sprout into that of the ordinary arm as well as stimulating it to grow by one more site. The sprout finisher '6' also changes a following gene '4' into '7', which makes the final growth of the sprout.

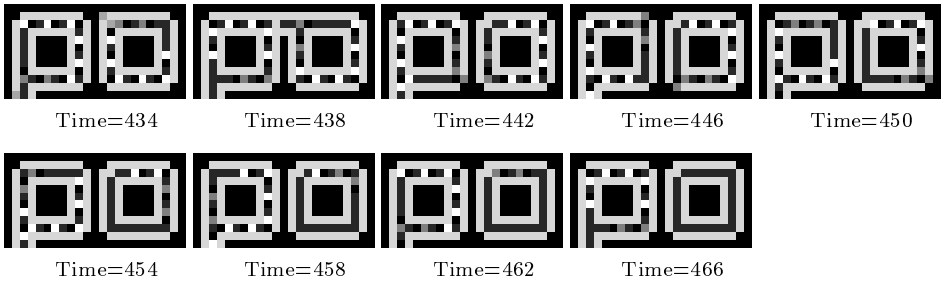


Figure 4.11: Death as functional failure of the SR loop. The right loop tries to thrust its arm to the left (time=438). However, since the left area is already occupied by another loop, a sheath fragment is generated on the pathway of genes at the upper left part of the right loop (time=442). Genes traveling in the loop are absorbed one after another by the generated fragment (time=446–462). Eventually, all the genes have been extinguished, and only a circular tube filled with cores are left (time=466).

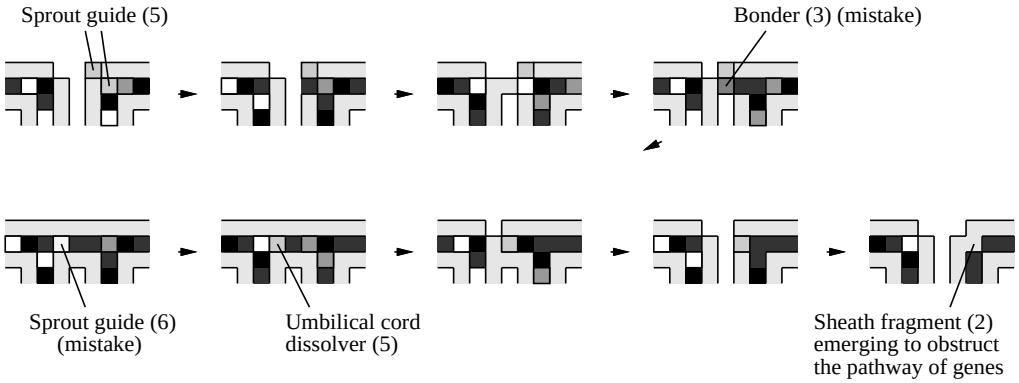


Figure 4.12: Mechanism of the beginning of death of the SR loop. When the right loop tries to thrust its arm to the left, a bonder '3' and a sprout guide '6' are generated successively by mistake. Then, the sprout guide '6' turns into an umbilical cord dissolver '5', which goes back and dissolves the connection between the two loops. Eventually it turns into a sheath fragment '2' in the pathway of genes to obstruct their propagation.

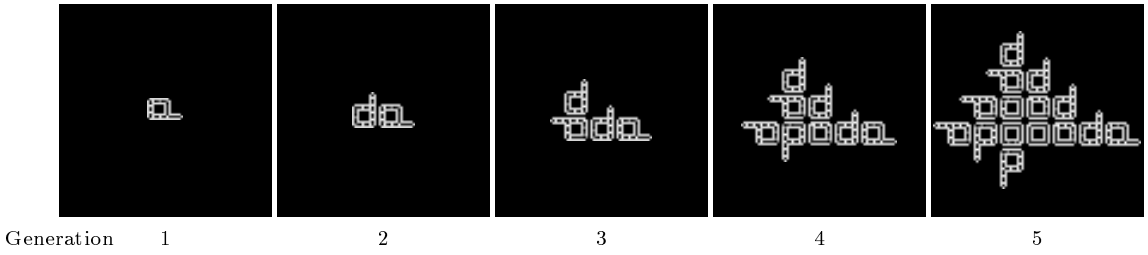


Figure 4.13: Growth of the colony of the SR loop. One generation takes 151 CA updates, which the loop needs for one complete self-reproduction. The colony grows as time proceeds, and its inner area becomes filled with dead loops like a colony of coral.

4.5 Experimental tools used in this study

At the end of this chapter, the concrete experimental tools we used in this study are mentioned briefly.

Although we anticipate the future implementation of the model on hardware, all the results reported in the following chapters have been obtained by computer simulations. We utilized mainly a DEC Digital Personal Workstation 500au for the principal platform of the simulations. The softwares for simulation/visualization of the CA are the original ones we developed for this study. These softwares (for Unix/MS-DOS machines) are available from the author's WWW page introduced in section 1.2. The editing of state-transition rules and initial configurations of the CA was conducted by using an ordinary text editor, which was quite a tedious task and still has considerable room for further improvement.

Chapter 5

A STRUCTURALLY DISSOLVABLE SELF-REPRODUCING LOOP

5.1 Introduction

The first step to advance Langton's SR loop to a truly evolving one is to introduce death as structural dissolution into the model in which death as functional failure is already featured. In this chapter, we introduced a new *dissolving state* '8' into the set of states of the CA, while exactly preserving the states '0'–'7' and all the state-transition rules proposed by Langton[1]. We expected that structural dissolution would be realized through propagation of this dissolving state along the tube of the loop. This improved loop is termed a "structurally dissolvable self-reproducing (SDSR) loop"[44, 47]. Through the experiments conducted with the SDSR loop, several interesting phenomena not previously reported in the world of SR loops are observed.

5.2 Implementation

5.2.1 Constructing state-transition rules

In this section, the details of implementing the state-transition rules of the SDSR loop are described step by step.

First, we need to address a technical problem inherent in the rules published by Langton, as those were limited to an indispensable rule set for simple self-reproduction only, while many situations still remained undefined. We suppose that Langton implicitly assumed that the environment in which the loops would expand was infinite, so that the loops

[1]As we remark later, there is *one* exception to this almost complete preservation.

would not encounter any irregular situation not previously defined in the state-transition rules. However, if space is designed to be finite by imposing periodic boundary conditions at the edges, it is obvious that the loops will eventually face the limitation of space and the appearance of irregular situations will cause their activities to halt. To avoid this inconvenience, before introducing the possibility of structural dissolution, the state-transition rules needed to be extended to redefine rules which had been left undefined in Langton's SR loop, by a natural extension of the ideas of Codd and Langton[11, 27] as follows:

1. A background '0' in the tube next to a core '1' will turn to '1'. All other '0's will remain as is.

2. A core '1' in the tube next to a gene '7' will turn to '7'. Or, a '1' in the tube next to a messenger '6' will turn to '6'. Or, a '1' in the tube next to a gene '4' will turn to '4'.

3. Signal '4', '6' and '7' in the tube next to '0' will turn to '0'.

4. A sheath '2' next to a bonder '3' will turn to '1'. Or, a '2' next to another '2' will remain as is.

Extensions 1, 2 and 3 define general rules for propagation of genes and messengers in the tube, and extension 4 defines general rules for the connection of two tubes. The criterion for judgment whether a site is in the tube or not was taken to be whether at least two sites in the state '1', '2', '4', '6' or '7' are included in the four sites neighboring itself.

Second, a new *dissolving state* '8' was introduced into the set of states of the CA according to the strategy we planned in chapter 4 with the following state-transition rules:

1. A dissolving state '8' will unconditionally turn to '0'.

2. When neighboring an '8',

   (a) both a background and a core ('0' and '1') will turn to '8' if there is a sheath or a signal in the four sites next to themselves. This rule represents contagion of the dissolving state. Otherwise, they will remain unchanged.

   (b) a sheath and some signals ('2', '3' and '5') will turn to '0'. This rule represents dissolution of tube structure.

   (c) the remaining signals ('4', '6' and '7') will turn to '1'. This rule represents dissolution of signals in the tube.

Owing to these rules, the dissolving state acquires the ability to travel along the tube and dissolve a neighboring tube structure. The direction of propagation of a dissolving

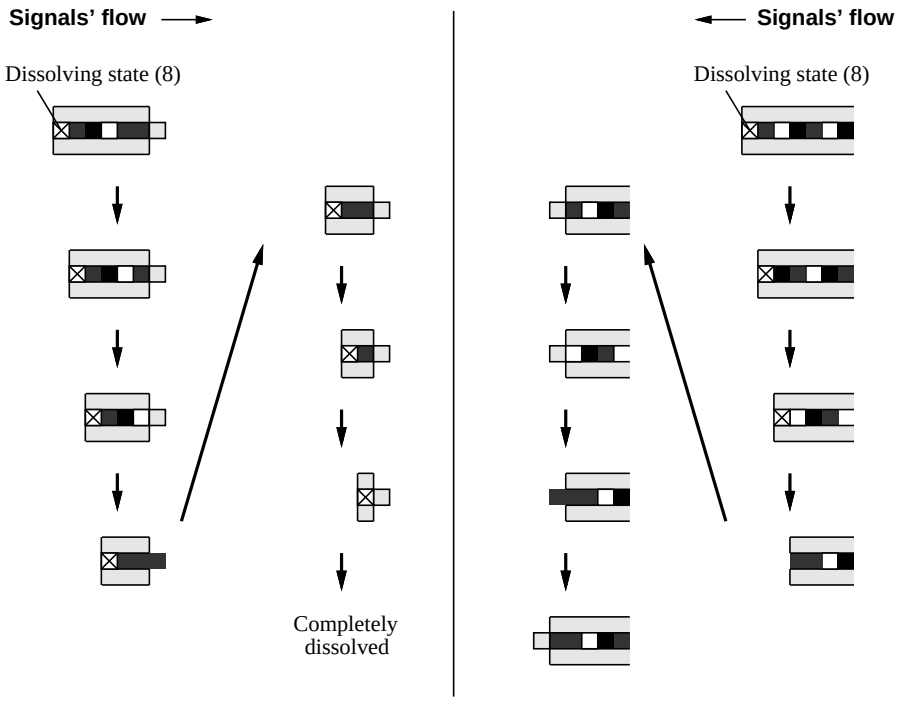


Figure 5.1: Behavior of dissolving state '8'. When a dissolving state travels in the same direction as the signals flow (left), it can completely dissolve the structure to the tip of the tube. However, when a dissolving state travels against the direction of flow (right), it is blocked by the sequence of signals. For these reasons, a dissolving state will extinguish the tube structure along the same direction as the signals' flow.

state is the same as that of the signals' flow in the tube, because, if it travels against the signals' flow, it is blocked by the sequence of signals. This behavior of the dissolving state is shown in Figure 5.1. This nature of the dissolving state leads to the possibility of partial structural dissolution (i.e., dissolution of only a part of the arm). The effect caused by such partial dissolution is mentioned later.

Third, only *one* alteration was made in the state-transition rules inherited from Langton as follows:

$$\frac{\texttt{CTRBL->I}}{\texttt{11152->2}} \longrightarrow \frac{\texttt{CTRBL->I}}{\texttt{11152->8}}$$

Here, `CTRBL` and `I` stand for the states of sites in the neighborhood template and the image of state transition, respectively (see also instructions in appendix A). This alteration implies that the SDSR loop will generate a dissolving state in itself for the situation in which the SR loop will generate a sheath fragment in itself at the beginning of death process (Fig. 5.2).

Fourth, finally, it was decided that all the situations left undefined after the above

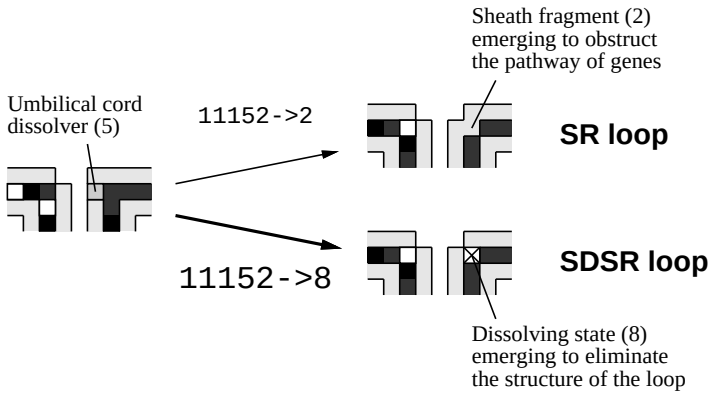


Figure 5.2: Only one alteration in the state-transition rules in implementing the SDSR loop. Owing to this alteration, the SDSR loop will generate a dissolving state in its body at the beginning of death process.

definitions are implemented are uniformly directed to the dissolving state '8'.

Via these definitions, once a site takes on the dissolving state, a contiguous structure which includes that site will be extinguished quickly by the dissolving state traveling in the tube. Thus, the SDSR loop acquires the capability of structural dissolution in addition to self-reproduction. The procedure of the above implementation is also summarized in appendix B.

5.2.2 Fundamental behaviors

The SDSR loop designed here can dissolve itself in the manner depicted in Figure 5.3. This can be regarded as a phenomenon similar to programmed death of biological cells, such as apoptosis and programmed self-decomposition[35, 36]. However, it is important to point out the fact that the structural dissolution of the SDSR loop is programmed not in the genes of an organism, but in the state-transition rules, the natural laws of this CA world.

In addition, the SDSR loop also acquires some ability to overcome external difficulties. For example, when several obstacles are placed in front of the loop, it is possible that the loop eliminates them by dissolving the tip of its arm together with obstacles and reproduces itself correctly, as shown in Figure 5.4. This can be regarded as a phenomenon similar to the cutting of a lizard's tail, i.e., the action of disposing partial organs of an individual in order to overcome external difficulties and achieve survival of the individual. However, we should note here that this ability of the SDSR loop is derived from the manner of extension of the state-transition rules in which the direction of propagation of the dissolving state is limited to the same as that of signals in the tube. In other words,

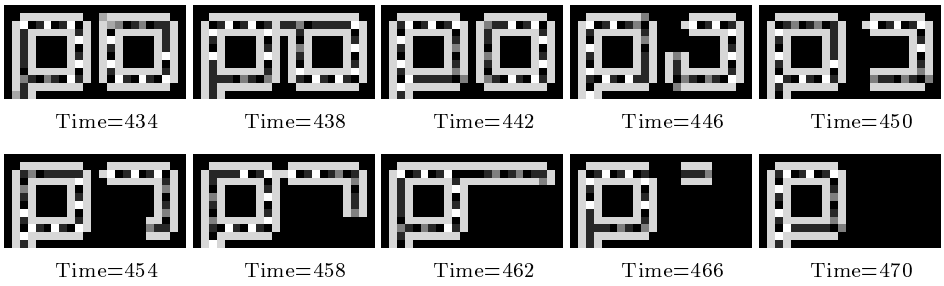


Figure 5.3: Death as structural dissolution of the SDSR loop. The right loop tries to thrust its arm to the left (time=438). However, since the left area is already occupied by another loop, a dissolving state is generated on the pathway of signals at the upper left part of the right loop (time=442). The dissolving state travels counterclockwise along the tube and dissolves neighboring sheaths one after another (time=446–462). Eventually, the structure of the right loop is completely dissolved (time=466–470).

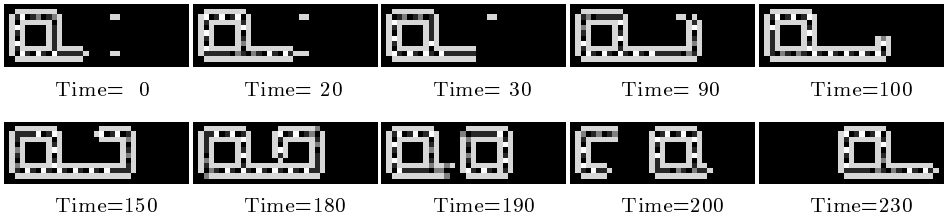


Figure 5.4: Partial structural dissolution of the SDSR loop. Two sheath fragments are placed in the area in which the loop intends to reproduce its offspring (time=0). As the loop thrusts an arm, a dissolving state appears because of collision of a tip of the arm with a sheath fragment. The dissolving state extinguishes the fragment together with the tip (time=20–100). After the loop has managed to eliminate obstacles in such a way, it reproduces the offspring correctly into the right area (time=150–190). However, since the length of the arm is changed in the process of elimination of fragments, a situation inconsistent with the mechanism of the loop has emerged in the parent, so its structure is consequently dissolved (time=190–230).

this ability is induced not by the introduction of structural dissolution itself, but by the fact that the partial dissolution of an individual is enabled at the next level.

5.3 Experiments

In this section, several experiments using SR loops and SDSR loops are reported. The SR loop is provided with the same extension of state transition rules as the SDSR loop except for the rules concerned with the dissolving state '8'. All sites under undefined situations in the SR loop are set to remain with no state transition.

5.3.1 Self-reproduction in infinite space

The first experiment is to breed loops in infinite space. Figure 5.5 shows the temporal development of spatial distribution of SR loops and SDSR loops in infinite space. One generation takes 151 CA updates, which the loop needs for one complete self-reproduction.

The behavior of SDSR loops at the edge of the colony is identical to that of SR loops at the same location, because there is no deficit in space so that no dissolving states appear in the SDSR loops at that point. However, at the central area of the colony, it is observed that many SDSR loops dissolve their own structure and provide new space for neighbor loops to reproduce themselves, while corpses of SR loops which died because of functional failure remain at the same location in the world of SR loops. Thus the colony of SDSR loops looks more sparse than that of SR loops.

Figure 5.6 shows the temporal development of the number of living individuals in the process of self-reproduction of both SR and SDSR loops. We can see from this figure that the number of living SDSR loops is much larger than that of living SR loops. Since SDSR loops can reproduce themselves into the central area of the colony (which is filled with corpses in the world of SR loops), they arrange themselves into the shape of a collapsing spiral. We can therefore estimate that the number of living individuals of SDSR loops approaches $\mathcal{O}(\text{Generation}^2)$ as time proceeds, substantially different from the number of living individuals of SR loops, which is estimated to be $\mathcal{O}(\text{Generation})$ as they can reproduce only at the edge of the colony.

5.3.2 Self-reproduction in finite space

The second experiment is to breed loops in *finite* space. Under this condition, the advantage of SDSR loops becomes even more apparent than in the former experiment. Figure 5.7 shows the development of spatial distribution of SR loops and SDSR loops in finite space. The size of the space is restricted to 200×200 sites, where periodic boundary conditions have been applied in both x and y directions.

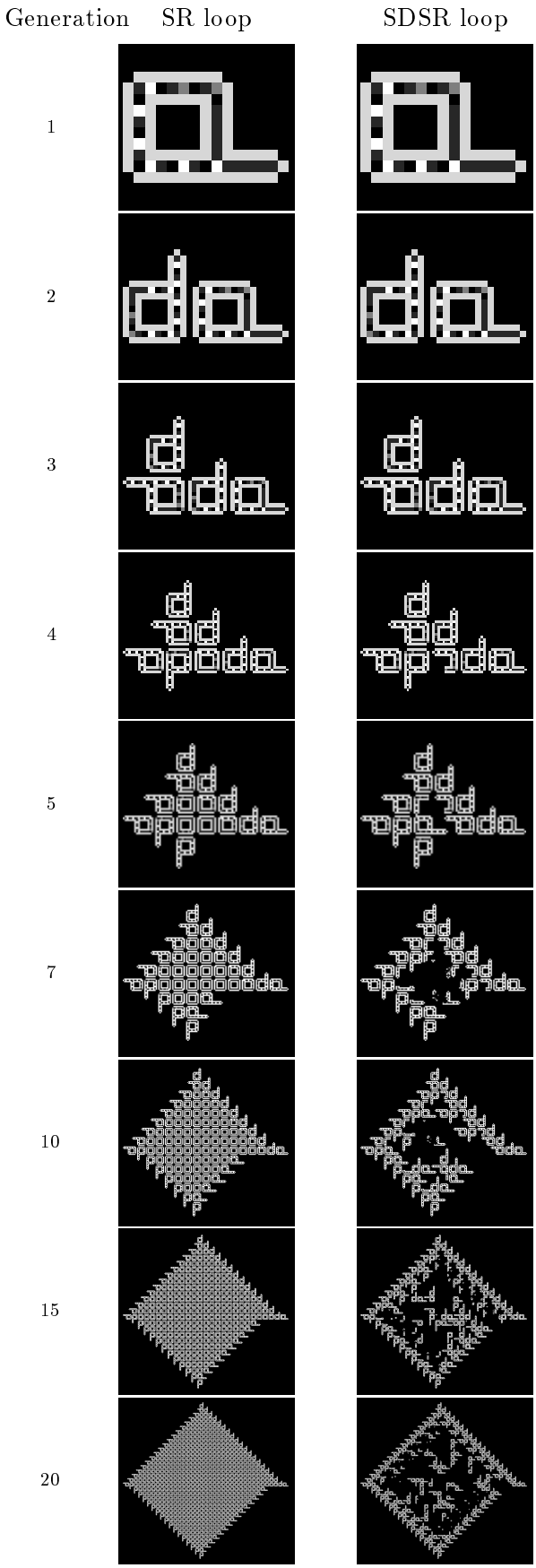


Figure 5.5: Temporal development of spatial distribution of SR/SDSR loops in infinite space. Each picture is scaled differently to the size of the colony.

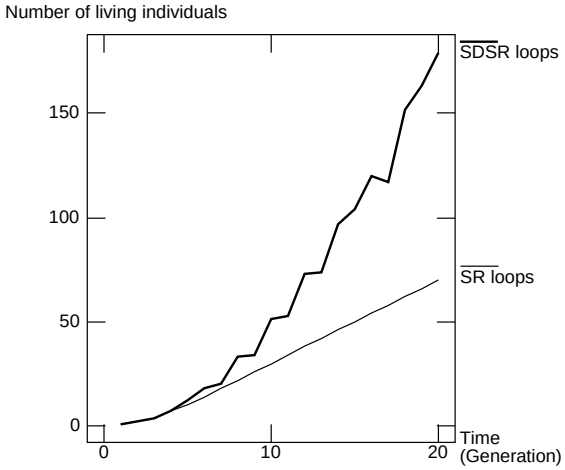


Figure 5.6: Temporal development of the number of living individuals of SR/SDSR loops in infinite space. The number of living SDSR loops is much larger than that of living SR loops because the SDSR loops can reproduce themselves into the central area of the colony which is filled with corpses in the world of SR loops.

In the world of SR loops, when the opposite edges of the colony collide with each other due to the periodicity of space, the irregularity of the emerging situations leads to a failure of the loops' function, so that their self-reproductive behavior halts. Eventually, the entire space is filled with static patterns including corpses of loops. After that, no changes can happen in this world any more. On the other hand, in the SDSR loop world, as new regions which can be used for self-reproduction of new loops are continuously produced by structural dissolution, the self-reproduction of loops is actively maintained. Although SDSR loops have a probability of accidental extinction of all living individuals caused by structural dissolution, they continue self-reproduction for a significantly longer period of time, at least ten thousand generations in this experiment. Thus, there can be no doubt that SDSR loops have attained a semi-permanent state of dynamic stability, while SR loops tend to fall into a static non-living state.

In addition, we observed a number of merged loops shown in Figure 5.8, which were produced in the course of self-reproduction of SDSR loops. Because SDSR loops can continue their self-reproductive behavior for a long period owing to structural dissolution, they are faced occasionally with rare conditions which can produce these merged loops by direct interaction of phenotypes. However, since they are generated by bonding arms of more than one loop, they have lost their construction arms which are necessary for self-reproduction. In this sense, they can be regarded as corpses having lost the capability of self-reproduction because of functional failure. They are all destined to be eliminated from the space sooner or later due to the structural dissolution caused by the attack from

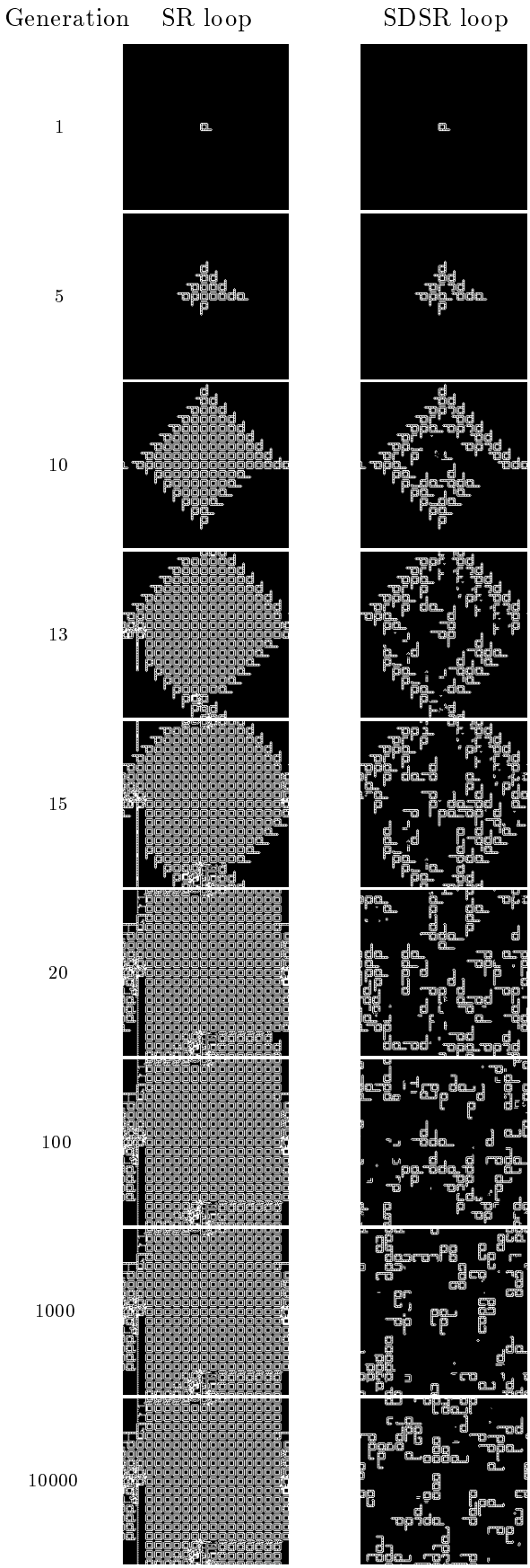


Figure 5.7: Temporal development of spatial distribution of SR/SDSR loops in finite space. Each picture is scaled to 200×200 sites.

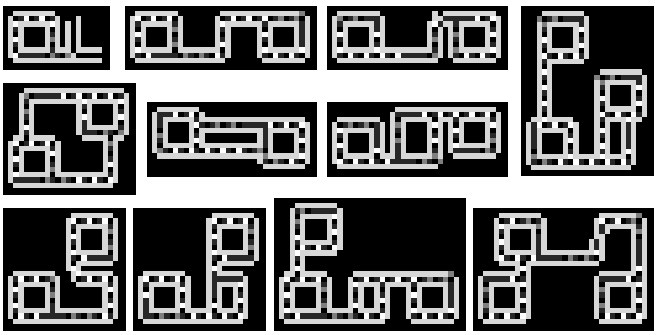


Figure 5.8: Examples of merged loops produced by direct interaction of phenotypes in the course of self-reproduction of SDSR loops.

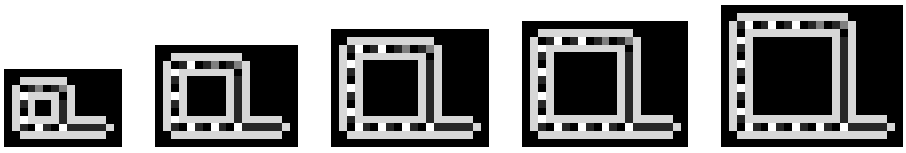


Figure 5.9: Examples of loops of different species. In this dissertation, species n is defined by $\mathcal{S}_n$ which is a set of structures that contain a square tube covered by sheath state '2's whose inner edges are n sites long. According to this definition, the normal loop of species n contains n gene '7's.

other loops living in their vicinity.

5.3.3 Struggle for existence in finite space

The third experiment is to observe the struggle for existence among SDSR loops of different species in finite space. As mentioned in section 3.3, the definition of species of artificial organisms in CA is subject to change according to the sets of structures used for the definition, thus we should clarify what kind of sets we use here. In this dissertation, we define the species of the loop by letting $\mathcal{S}_n$ (a set of structures of species n) consist of structures that contain a square tube covered by sheath state '2's whose inner edges are n sites long. According to this definition, the normal loop of species n contains n gene '7's.

Though Langton's loop contains exactly six gene '7's, we can construct loops of arbitrary species using extended state-transition rules, as long as not less than four gene '7's are contained in the loop. Figure 5.9 shows examples of loops of different species.

When loops of two different species were placed and bred in the same finite space, we observed that in the SDSR loop world one species drove the other into extinction mostly in less than ten thousand updates. Figure 5.10 shows an example of competition between

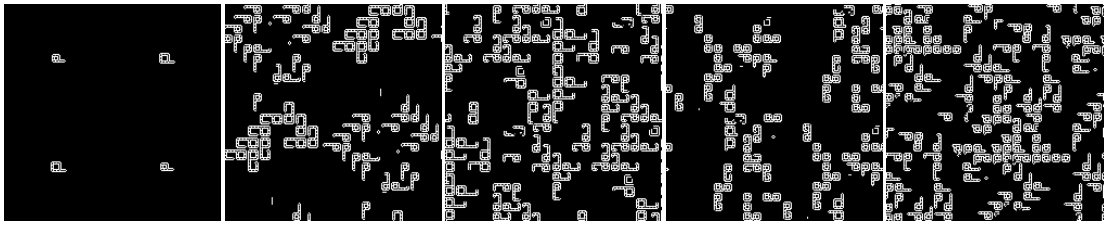


Figure 5.10: Struggle for existence between SDSR loops of species 4 and 6. At the beginning, loops of species 4 are placed in the upper left and lower right regions of the space, and loops of species 6 in the upper right and lower left regions (time=0). As time proceeds, loops of species 4 show that they are superior to those of species 6 (time=1000–3000). Eventually all loops of species 6 are exterminated (time=4000). The space is of 200×200 sites.

species 4 and 6. The records of a round-robin among species from 4 to 12 are listed in Table 5.1. It is observed that the small species generally fit this world better than the large ones. This is a trivial result as the smaller the size of loops, the faster they can reproduce themselves (as the reproduction time is shorter), with less detriment due to structural dissolution of one individual for the entire colony.

In addition, it was also observed that the species of $3n$ such as 9 and 12 displayed relatively high fitness, which seems peculiar to the model used this time. For example, the process that species 9 exterminates species 7 is shown in Figure 5.11. The reason for this phenomenon is probably a combination of two facts: One is that the original state-transition rules by Langton were designed to be optimum when applied to loops of species 6, and the other is that some situations which occur in a loop of species m are likely to occur in the same way to a loop of species $m \pm 3$ because every signal traveling in the loop is comprised of three sites of the CA. Thus, in the strict sense, it can be said that the species $6 + 3i$ $(i = 0, 1, 2, ...)$ have relatively high fitness in this world.

The fact that both the alteration of phenotypes caused by their direct interaction and the competitive exclusion acting between loops of different species are observed in this world suggests that the SDSR loop is potentially evolvable with variations caused by direct interaction of phenotypes.

However, unfortunately, such direct interaction of phenotypes of SDSR loops did not produce any new variety of living loops in this stage. This is the very problem we are going to resolve in the next chapter.

Table 5.1: Records of a round-robin among the SDSR loops of species from 4 to 12. The species that exterminated the other was considered as the winner of that match. Each pair of species were examined three times in three spaces of different sizes ($200 \times 200, 202 \times 202, 204 \times 204$ sites) in order to prevent the result from being influenced strongly by a slight difference in the initial configurations. 'O' in the table stands for three wins and no defeat, 'o' for two wins and one defeat, 'x' for one win and two defeats, and 'X' for no win and three defeats, respectively, in terms of the species A. According to the results listed in this table, the ranking of the species is 4, 5, 6, 7, 9, 8, 12, 10, 11, in the order of fitness.

A	B								
	4	5	6	7	8	9	10	11	12
4		O	O	O	O	O	O	O	O
5	X		O	O	O	O	O	O	O
6	X	X		O	O	O	O	O	O
7	X	X	X		o	o	O	O	O
8	X	X	X	x		X	O	O	O
9	X	X	X	x	O		O	O	O
10	X	X	X	X	X	X		O	x
11	X	X	X	X	X	X	X		X
12	X	X	X	X	X	X	o	O	

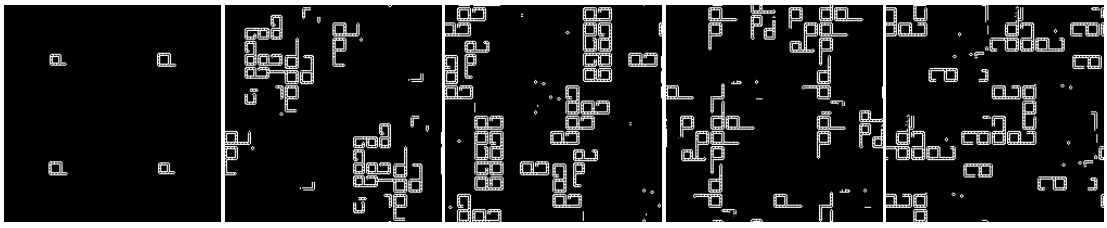


Figure 5.11: Struggle for existence between SDSR loops of species 7 and 9. At the beginning, loops of species 7 are placed in the upper left and lower right regions of the space, and loops of species 9 in the upper right and lower left regions (time=0). In this case, species 9, which is seemingly disadvantageous because of its larger size, eventually exterminated species 7 after a long close combat (time=8000). The space is of 200×200 sites.

Chapter 6

EVOLOOP: AN EVOLVING SDSR LOOP

6.1 Introduction

Natural selection acting among different kinds of loops was realized in the previous chapter through the implementation of SDSR loops. We also found some phenomena similar to variation in the SDSR loop world. Then, the next step toward the realization of evolution of self-reproducing loops is to introduce the actual variation to the model.

The approach having been taken in the former artificial evolutionary systems in order for the introduction of variation has usually been to fluctuate the genomes of artificial organisms explicitly by using pseudo-random numbers. On the other hand, many variants have been produced in the previous experiment using SDSR loops that was conducted without any stochastic operation. So, if we could make them stay and proliferate in the population, it would become possible to simulate all the essentials of evolution within a completely deterministic system, which would greatly facilitate the simplification of the simulation algorithm and its future implementation on the hardware. Therefore, it was decided that no stochastic mutation would be explicitly used in this study; to be concrete, the introduction of variation was achieved in implicit ways—through enhancing the adaptability of the self-reproductive mechanism of the SDSR loop and slightly modifying its initial structure.

In this chapter, a new evolutionary system *evoloop* (an evolving SDSR loop)[45, 46, 48] is implemented, and its characteristic behaviors are observed.

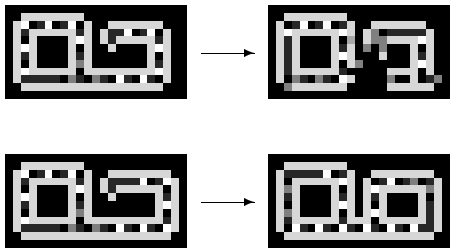


Figure 6.1: What happens if the form of the arm of the SDSR loop is altered by force during self-reproduction. In the upper case, structural dissolution occurs, while in the lower case, the loop falls into a sterile structure. The SDSR loop cannot reproduce any self-reproductive offspring in such cases.

6.2 Implementation

6.2.1 Problem in the state-transition rules of the SDSR loop

The SDSR loop introduced in the previous section did not show any apparent evolvability. The reason why it could not actually evolve is that its state-transition rules which designated all the mechanisms necessary for self-reproduction and other behaviors were specialized only for a set of particular situations which appeared in the ordinary self-reproductive process of the original SR loop. Above all, the functions of signal '3', '5' and '6' that were strongly related to the germination of a new arm were especially dependent on the situations of their vicinity; thus, without more extension of the state-transition rules concerning these signals, even a slight fluctuation occurring to the situation such as a one-site discrepancy in propagation of signals could easily ruin the self-reproductive process of the loop.

The CA space discussed here involves no stochastic operation like random mutation of genotype. Therefore, if the evolutionary process of loops occurred in such a deterministic universe, it should be driven by the direct alteration of phenotypes of offspring loops during self-reproductive processes. However, the SDSR loop cannot reproduce any self-reproductive offspring when the form of its arm is altered by force during its self-reproductive process, as shown in Figure 6.1; it either generates a dissolving state or falls into a sterile structure. In such cases, neither the connection of the tip of the arm and its root nor the dissolution of the umbilical cord between the parent and the offspring occurs correctly, because in these cases the location of genes near the bonding T-junction is different from the situations expected by Langton's state-transition rules. Such rigidness of the rules seems to have prohibited the SDSR loop's evolution.

6.2.2 Reconstructing the state-transition rules

For resolution of the problem of the SDSR loop mentioned in the above section, it was necessary to make the self-reproductive mechanism described by the state-transition rules to be more "adaptable". The word "adaptability" used here intuitively means a degree of the variety of situations in which the structures in the CA space can retain their regular operations.

To enhance the adaptability of the self-reproductive mechanism of the SDSR loop, we reconstructed its state-transition rules carefully while keeping the fundamental behavior of the signals as is. We first defined the general rules concerned with sustenance of sheath structures and propagation of genes. Next, in order to clarify the behaviors which must be realized in the state-transition rules for self-reproduction, we divided the self-reproductive process of the SDSR loop into the following six phases:

1. to lengthen the construction arm,
2. to turn the tip of the arm left,
3. to bond the tip and the root of the arm together,
4. to dissolve the umbilical cord between parent and offspring,
5. to germinate a new sprout of the arm, and
6. to lengthen the new sprout of the arm.

Then, we refined each part of the state-transition rules relevant to each of the above six phases to make it adaptable to a greater variety of situations than before.

On granting the adaptability to the self-reproductive mechanism of the SDSR loop, some inadvertent complication of the old state-transition rules became a nuisance. To be concrete, in the CA of the SR/SDSR loops, the rules concerned with bonding of the tip and the root of the arm and germination of a new sprout of the arm in the parent loop were constructed in such a heuristic way completely dependent on some specific situations that they defied any modification (Fig. 4.8, 4.9). Ideally, the former mechanism should be improved to be able to bond the tip and the root whenever and wherever they meet, and the latter should be much simpler like the germination of a sprout in the offspring (Fig. 4.10). Furthermore, since the old rules had some redundancy that the place of a new sprout of the parent's arm was pointed by a messenger '5' traveling on the sheath while that of the offspring's was pointed by a different messenger '6' traveling in the tube, these two mechanisms should be integrated into the identical one.

To resolve these problems, we reconstructed the mechanisms of self-reproduction of the loop thoroughly. For example, the mechanism for germination and growth of a new sprout

was planned to be the identical one in both parent and offspring, in which the growth of the sprout is explicitly stimulated by genes. In order to equalize the length of the parent's sprout with that of the offspring's, we expected that all the gene '7's contained in the loop should be utilized for the growth of the sprout in any case. As a result, the length of the umbilical cord became longer than that in the SR/SDSR loop. For the dissolution of such a lengthened umbilical cord, signal '6' was re-assigned to be a special "umbilical cord dissolver" much more powerful than that in the old rules. The functions formerly possessed by signal '6' were re-assigned to '3', '4' and '5' in the new rules. The new functions of the states in this improved CA are listed in Table 6.1. Due to these re-assignments, the state-transition rules became more feasible for modification, thus we could easily refine the rules to make them keep their regular operations in a greater variety of situations.

Figure 6.2 depicts the general behaviors of the refined phases. The mechanisms concerned with phase 3, 4, 5 and 6 are reconstructed this time from scratch, while those concerned with 1 and 2 are exactly same as in the SR/SDSR loops. The detailed mechanisms of phase 3–6 are depicted in Figure 6.3–6.6.

After these operations for enhancing the adaptability of the self-reproductive mechanism, the dissolving state '8' was introduced into the set of states of the CA in the same way as in the SDSR loop. Note that the extension of the rules which had been conducted at the beginning of the implementation of the SDSR loop is no longer necessary here, since all the possible situations which may happen in the self-reproductive process of the improved loop has been defined thoroughly by the above-mentioned enhancement of the adaptability of the self-reproductive mechanism.

We eventually obtained a new loop which was extremely resistant to the fluctuation of environmental conditions with neither increase in number of both states and neighborhood sites of the CA nor alteration of the structure of the loop. We named this "*evoloop*". How to construct the complete state-transition rule set of the *evoloop* is shown in appendix C.

6.2.3 Fundamental behaviors

Self-reproduction of the *evoloop* is shown in Figure 6.7. Since the growth of the sprout of the *evoloop* is explicitly stimulated by gene '7's contained in its body, the length of its umbilical cord is longer than that of the SR/SDSR loop. Thus, the colony of the *evoloop* looks a little sparser than that of the SR/SDSR loop.

It is remarkable that, owing to the adaptability enhanced above, some intriguing interactions of loops emerge in the *evoloop* world, which have never occurred in the SDSR loop world. Figure 6.8 shows, for example, a takeover of the arm happening between two *evoloops*. In this case, the right loop takes over the arm of the left loop, and consequently a small rectangular variant is produced between the two loops. Due to the high adaptability of their self-reproductive mechanism, the two parent loops as well as the produced variant

Table 6.1: Names and functions of the states in the improved CA.

Fundamental elements

State	Name	Functions
'0'	Background	The quiescent state of this CA.
'1'	Core	To fill the tube of the loop and conduct genes in it.
'2'	Sheath	To form the tube structure of the loop.

Signals

State	Name	Functions
'3'	Left indicator	To support left turning of the arm.
	Bonder	To support bonding of the two arms.
	Sprout generator	To support germinating of the new sprout.
	Sprout capper	To cap the tip of the sprout.
	Sprout finisher	To finish growth of the sprout.
'4'	Gene	To keep genetic information of left turning of the arm and finishing of growth of the sprout.
	Sprout guide	To support growth of the sprout.
'5'	Messenger	To point where the loop should germinate a new sprout.
'6'	Umbilical cord dissolver	To dissolve the umbilical cord between parent and offspring.
'7'	Gene	To keep genetic information of straight growth of the arm and the sprout.

Dissolving state

State	Name	Functions
'8'	Dissolving state	To dissolve a contiguous structure of the loop.

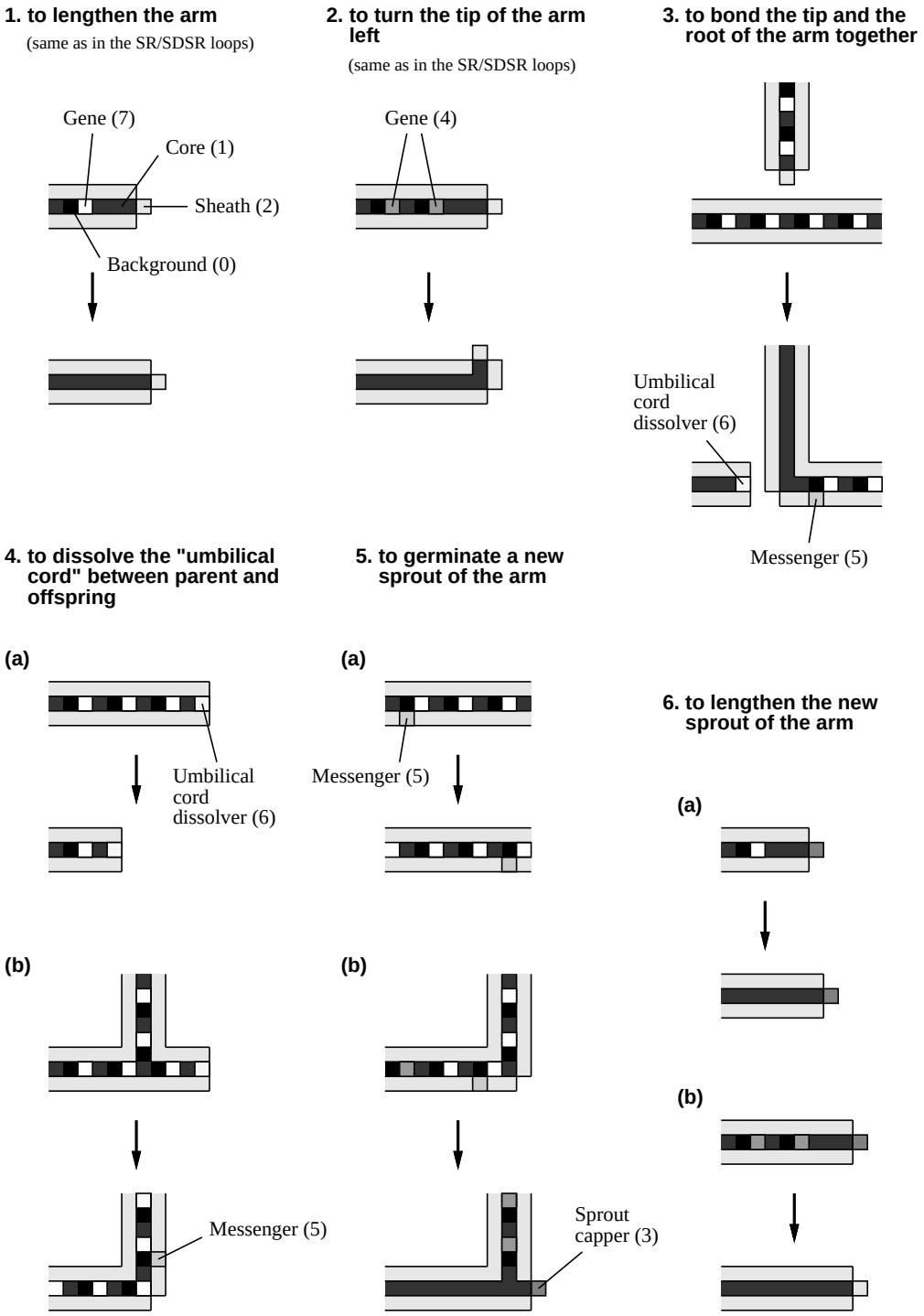


Figure 6.2: General behaviors of six phases of the self-reproductive process of the improved loop. The mechanisms concerned with phase 1 and 2 are exactly same as those in the SR/SDSR loops (Fig. 4.6, 4.7), while those concerned with 3, 4, 5 and 6 are reconstructed from scratch (Fig. 6.3–6.6).

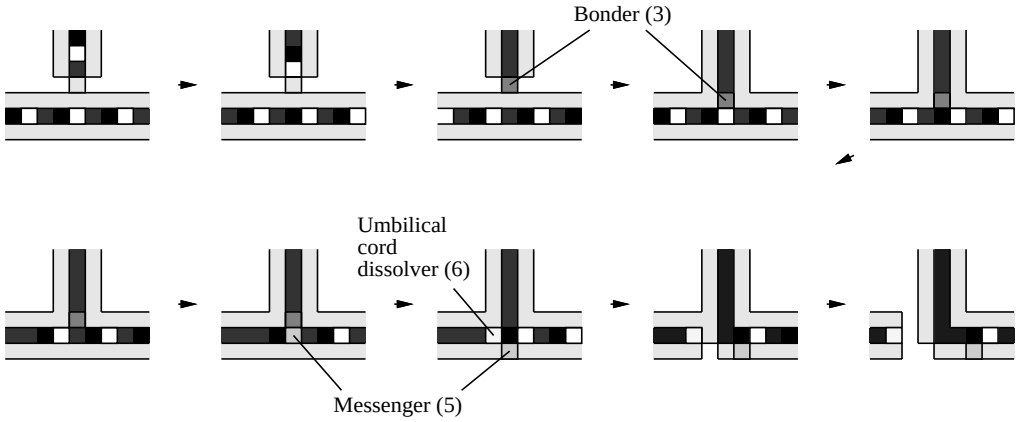


Figure 6.3: Mechanism of the phase to bond the tip and the root of the arm together in the improved loop. When a gene '7' hits the tip touching the middle of the arm, the sheath '2' having capped the tip turns into a bonder '3', while the gene '7' disappears. This bonder '3' moves onto the lower sheath and then begins to wait there for another gene '7' coming from the parent loop. When a gene '7' comes into the junction point with the bonder '3' waiting, it alters into a messenger '5'. Then, next, the bonder '3' disappears, the gene '7' recovers from the messenger '5', an umbilical cord dissolver '6' appears out of the messenger '5', and the messenger itself goes through onto the outside sheath. The messenger '5' begins to travel along the sheath structure being trailed by the gene '7' toward the next corner, while the umbilical cord dissolver '6' goes back toward the parent loop.

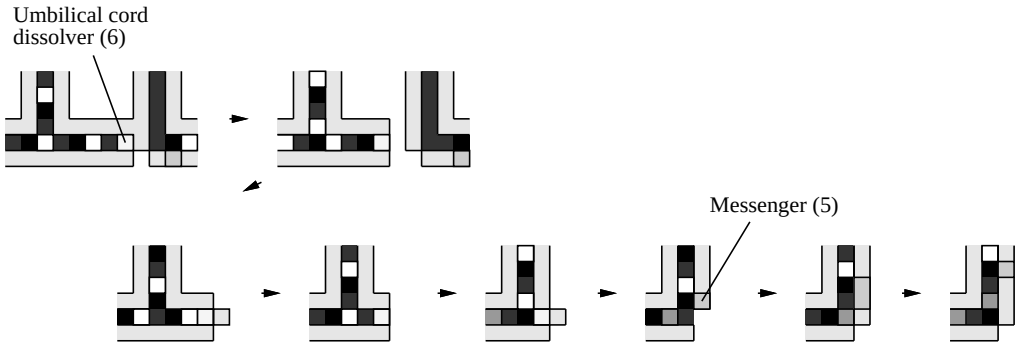


Figure 6.4: Mechanism of the phase to dissolve the umbilical cord between parent and offspring of the improved loop. The umbilical cord dissolver '6' breaks the connection between the cord and the offspring, and begins to travel backward against the genes' flow, dissolving structures of the cord. When the dissolver '6' arrives at the T-junction (i.e., the root of the cord) in the parent loop, it generates a messenger '5' onto the sheath of the loop and itself disappears. The corner (which was formerly the T-junction) broken by the dissolver '6' is quickly repaired, and the messenger '5' begins to travel along the sheath being trailed by a gene propagating in the tube.

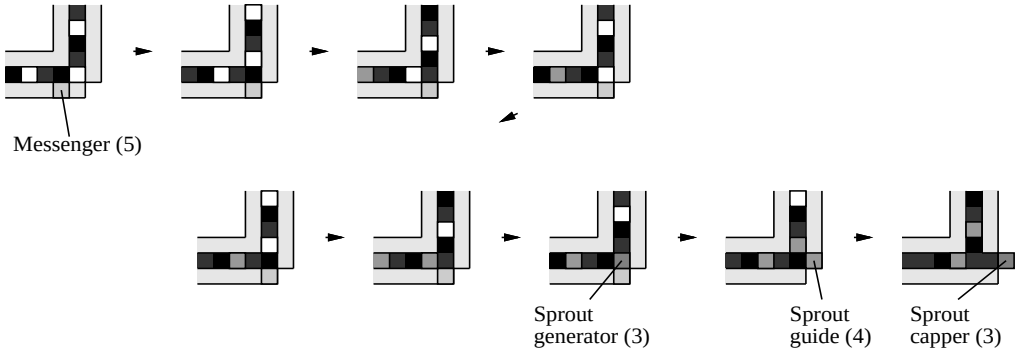


Figure 6.5: Mechanism of the phase to germinate a new sprout of the arm of the improved loop. When the messenger '5' having been generated in the above-mentioned phases arrives at the next corner, it begins to wait there for another gene '4' coming. When a gene '4' comes into the corner with the messenger waiting, it alters into a sprout generator '3'. Then, next, the gene '4' recovers from the sprout generator '3', a sprout guide '4' appears onto the outside sheath out of the sprout generator '3' , and the sprout generator itself and the messenger '5' disappear. The sprout guide '4' quickly turns into a new sprout of the arm capped by a sprout capper '3'.

can continue their self-reproductive activity after such an accident. We expected such a direct interaction of phenotypes of *evoloops* to drive their evolution.

We should notice, however, that the state-transition rules of the *evoloop* have *no* explicit mechanism for evolution; they are merely comprised of the phases necessary for self-reproduction of the loops.

6.2.4 Developing a method of observation

The *evoloop* world itself keeps no explicit information about the status of individuals/species or the interrelations between them, because all phenomena emerging in the space are generated only by local interactions of neighboring sites. So, a considerable amount of calculation is devoted to image processing in order to detect any meaningful information in the configuration.

In this study, we developed another binary (two-state) CA system specialized for image processing which automatically identifies the living loops in the space, and embedded this system into the simulator software. In order to simplify the computations involved in this process, we regarded $\mathcal{S}_n$ (a set of structures of species n) as consisting of all structures that contain an open square made of sheath '2' and signal '3' whose edges are n sites long. To be concrete, the configuration concerning only sheath '2' and signal '3' in the whole space is sent to this "observer" CA as a configuration of active sites, and then the observer CA applies the following state-transition rule to the configuration until it reaches the fixed point:

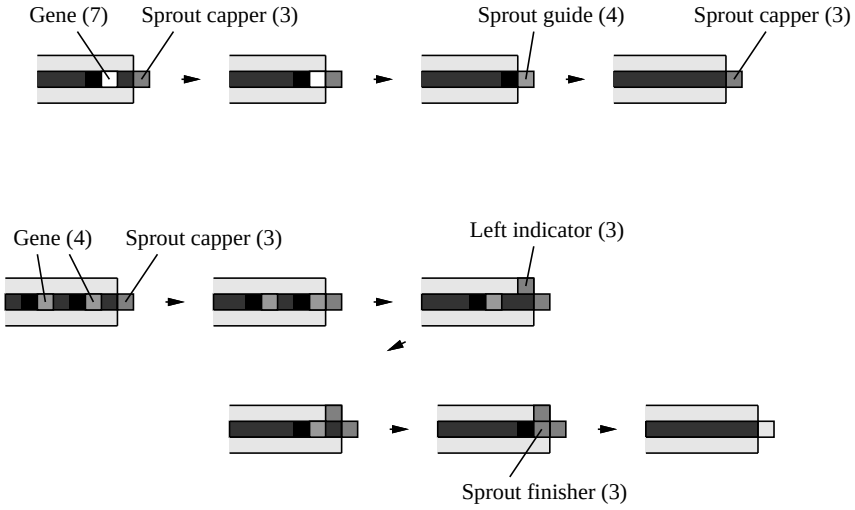


Figure 6.6: Mechanism of the phase to lengthen the new sprout of the arm of the improved loop. Upper: Growth of the sprout. This resembles the mechanism to lengthen the ordinary arm. When a gene '7' traveling in the tube structure hits the tip of the sprout, a sprout capper '3' having capped the tip turns into a sprout guide '4', and the gene '7' disappears. Then, the tip quickly becomes sheathed with a new sprout capper '3' and the sprout guide '4' disappears after one update; thus, the sprout becomes one site longer than before by a gene '7'. Lower: Finishing of growth of the sprout. When the first gene '4' traveling in the tube structure hits the tip of the sprout, a left indicator '3' emerges on the left-hand sheath of the tip, and the gene '4' disappears. This left indicator '3' remains as is until the second gene '4' comes. When the second '4' comes into the tip of the sprout with the left indicator '3', it alters into a sprout finisher '3'. Then, after one update, both of the left indicator '3' and the sprout capper '3' turns into the ordinary sheath, while the sprout finisher disappears. Consequently, the sprout is changed into the ordinary arm by two gene '4's. Since the sprout is germinated by a gene '4', and changed into the ordinary arm by a pair of gene '4's, the length of the sprout becomes equal to the total number of gene '7's sandwiched between them.

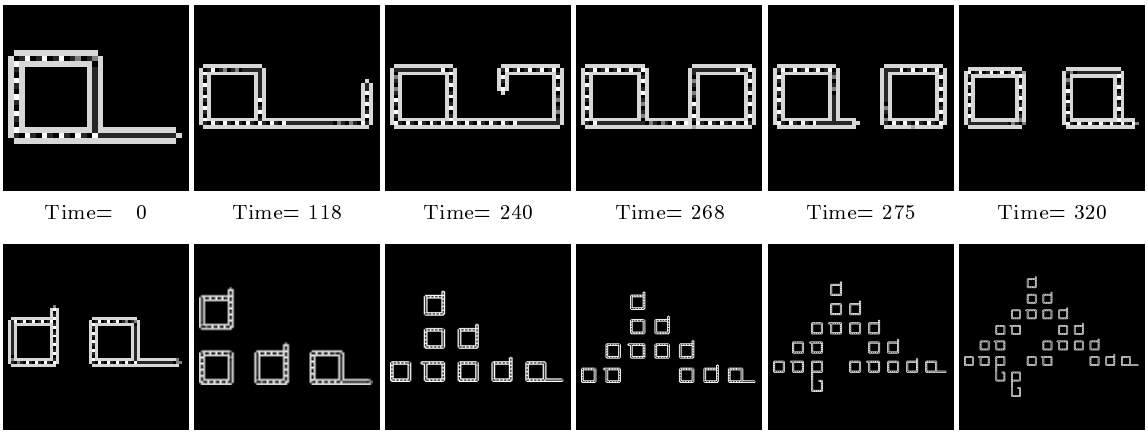


Figure 6.7: Self-reproduction of the *evoloop* of species 13. Note that the length of its umbilical cord is longer than that of the SR/SDSR loop, because the sprout of the arm of the *evoloop* is explicitly lengthened by all gene '7's contained in the loop.

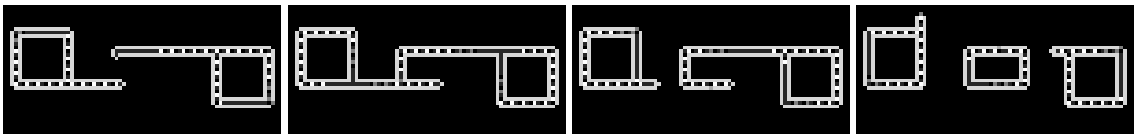
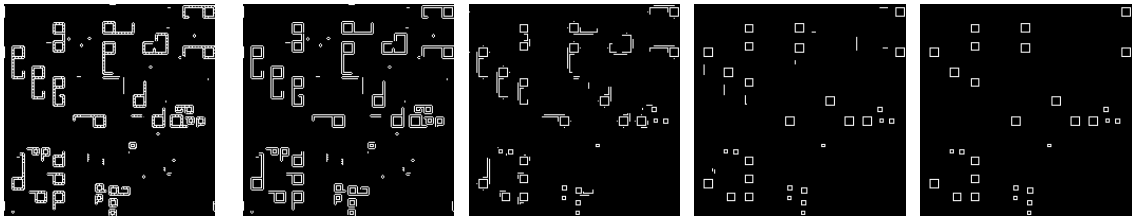


Figure 6.8: Takeover of the arm caused by the collision of two *evoloops*. The right loop takes over the arm of the left loop, and consequently a small rectangular variant is produced between the two loops. Due to the high adaptability of their self-reproductive mechanism, the two parent loops as well as the produced variant can continue their self-reproductive activity after such an accident.



Original image | Only '2' and '3' | 5 times processed | 15 times processed | Fixed point

Figure 6.9: Image processing by the "observer" CA for the automatic identification of living loops in the space. This observer CA receives the configuration concerning only sheath '2' and signal '3' in the whole space, and applies the rule for detecting only a circular structure with no branch to the configuration until it reaches the fixed point. Eventually, only the inner rectangular structures of the loops remain, and then the number and the size of squares are counted sequentially.

- The active site will remain if and only if neighboring exactly two other active sites, otherwise it will turn into a quiescent state.

Only a circular structure with no branch can survive this rule. In the *evoloop* world, such structures are only the inner rectangles of the loops, thus we can extract the information about the number and the size of living loops in the space by this method[1]. This detecting process is shown in Figure 6.9. After the configuration reaches the fixed point, the number and the size of squares are counted sequentially.

6.2.5 Evaluation of the adaptability

To compare the adaptability of the *evoloop* with that of the SDSR loop is important for the objective evaluation of the models. It is not appropriate for this purpose to statically calculate some quantity from only their state-transition rules, because the behavior of CA is strongly affected by an initial configuration as well as by state-transition rules. Rather, we should calculate the adaptability of the models dynamically from the actual result of explicit simulation of the behavior of the CA by using a specific initial configuration.

We need to discuss here how to measure the adaptability from the result of simulation. Though the notion of the adaptability has been used in this dissertation as a degree of the variety of situations in which the structures in the CA space can retain their regular operations, what the regular operation means here is not so obvious. However, in view of the strategy of this study in which irregular situations has been defined to generate dissolving states, it would be natural to define the regular operation of a structure as the continuation of the structure existing in the space without structural dissolution.

[1]We should note that this method still involves some errors because it cannot exclude dead structures which contain square sheaths. In addition, when a structure contains two or more squares in itself (e.g., a loop which has just closed its arm), it is mistaken for two structures by this method.

This definition suggests that the adaptability of the model can be roughly evaluated by measuring the average life expectancy (i.e., the average length of time from production till dissolution) of artificial organisms in the CA.

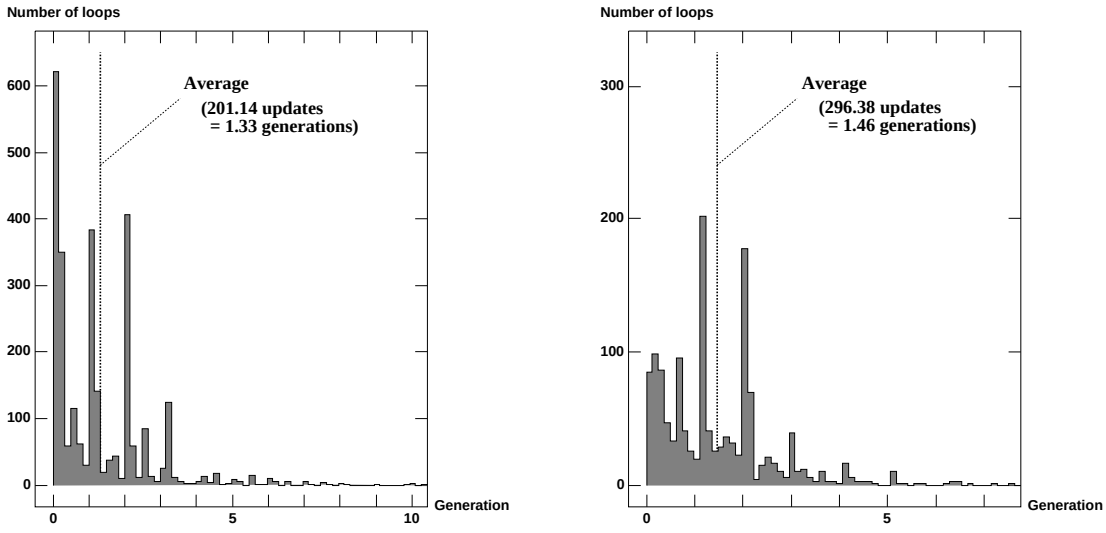
Based on this idea, we measured the lifetimes of all loops of species 6 which emerged during the first 10000 updates of the simulations of both SDSR loops and *evoloops* in the finite space of 200 × 200 sites. Initial configurations of both simulations were equally populated by an ancestral loop of species 6. The result is shown in Figure 6.10. These histograms indicate the distribution of lifetimes of the loops in both cases by accumulating them at intervals of 25 updates. We can find a clear difference between SDSR loops and *evoloops* in these histograms. In the SDSR loop world, the most lifetimes are of less than 25 updates which mean that these loops died immediately after their birth (Fig. 6.10: Left). On the other hand, in the *evoloop* world, the immediate death of loops after birth is not so many, and the most lifetimes are of 225–250 updates which mean that these loops died after they succeeded once in self-reproduction (Fig. 6.10: Right). Consequently, the average life expectancy of *evoloops* (296.38[update]/203[update/generation] = **1.46**[generation]) is longer than that of SDSR loops (201.04[update]/151[update/generation] = **1.33**[generation]). This result is considered to show quantitatively that the self-reproductive mechanism of the *evoloop* is more adaptable than that of the SDSR loop.

6.2.6 Preliminary experiments

In order to examine the evolvability of the *evoloop*, preliminary experiments of breeding *evoloops* of species 4–20 in finite spaces comprised of 99×99–401×401 sites were conducted. Each run was traced for 50000 updates.

The results are shown in Table 6.2. In these results it is indicated that the *evoloop* is actually more adaptable than before and has evolved to some extent. For example, in some cases the loop evolved to that of a larger species (Fig. 6.11: Left), and in other cases it generated some variants which lost their self-reproductive ability but became capable of reproducing smaller offsprings than themselves (Fig. 6.11: Right). Unfortunately, however, the evolution toward fitter species (self-reproductive smaller species) was not observed yet in these experiments.

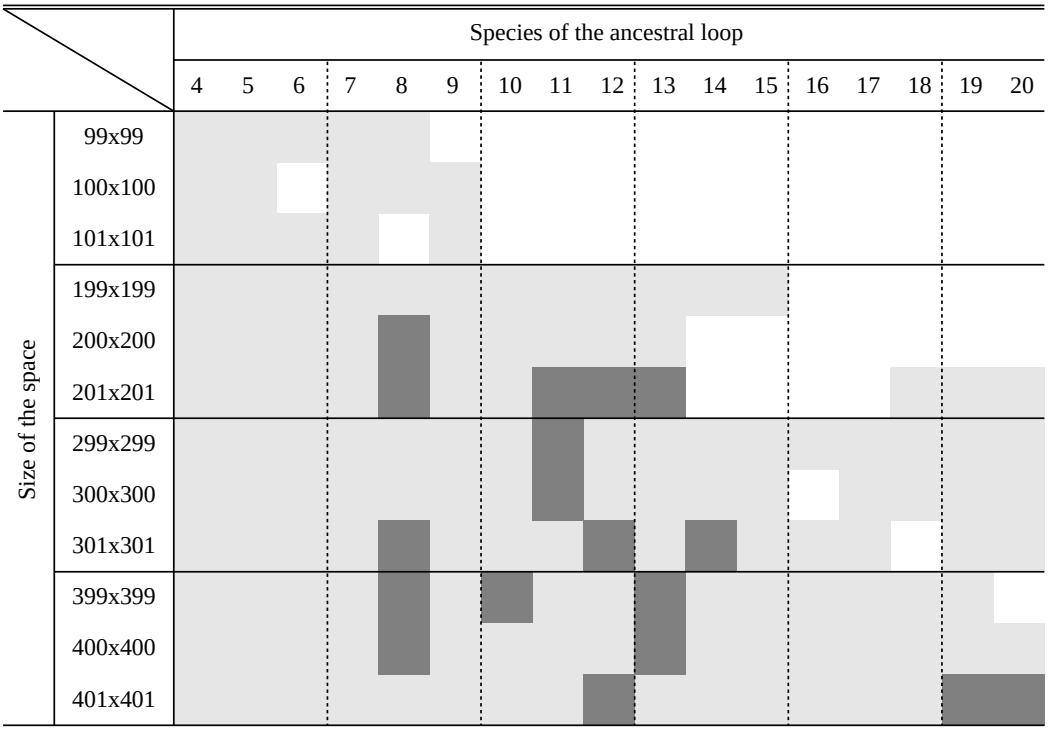
General behavior of the whole space of the *evoloop* world were also measured here. Figure 6.12 shows the averages of the numbers of active (non-zero) sites contained in the space and the standard deviations of their fluctuation calculated according to the results of these preliminary experiments. It is observed that the average number of active sites in the space decreases approximately in inverse proportion to the size of the species, while the standard deviation of the number of active sites are roughly constant independent of the species if the size of the space is fixed, and increases approximately in proportion to the

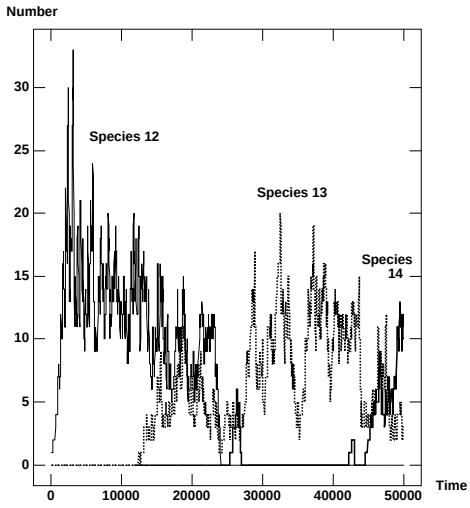


SDSR loops *evoloops*

Figure 6.10: Histograms of the distribution of lifetimes (lengths of time from production till dissolution) of SDSR loops (left) and *evoloops* (right) of species 6, being accumulated at intervals of 25 updates. Both simulations were traced for 10000 updates in the space of 200×200 sites with initial configurations populated by an ancestral loop of species 6. The lifetime of each loop having emerged in the simulation was measured by the length of continuation of its 6×6-site square detected by the observer CA mentioned in the previous section. These graphs are normalized so as to equalize the total area of grey regions which means the total number of the observed loops. In the SDSR loop world, the most lifetimes are of less than 25 updates which mean that these loops died immediately after their birth (left). On the other hand, in the *evoloop* world, the immediate death of loops after birth is not so many, and the most lifetimes are of 225–250 updates which mean that these loops died after they succeeded once in self-reproduction (right). Consequently, the average life expectancy of *evoloops* (296.38[update]/203[update/generation] = 1.46[generation]) is longer than that of SDSR loops (201.04[update]/151[update/generation] = 1.33[generation]).

Table 6.2: Results of preliminary experiments of breeding *evoloops* in finite spaces. The species of the ancestral loop ranges from 4 to 20, while the size of the space ranges from 99 × 99–401 × 401 sites. Gray indicates that the self-reproductive activity continued dynamically for 50000 updates in that case. Dark gray additionally indicates that the dominant species was changed to the larger species after 50000 updates in that case. Blank means that the whole system fell into a fixed point or a short-period limit cycle before 50000 updates.





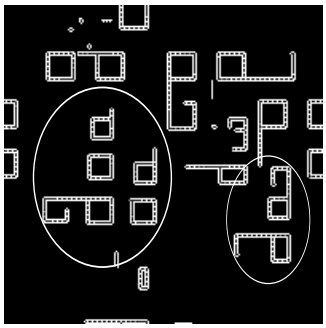


Figure 6.11: Evidences of the evolvability of *evoloops* observed in the preliminary experiments. Left: An example of evolution of the *evoloop* to larger species. Development of the numbers of living loops is shown. The ancestor is of species 12. The space is comprised of 201 × 201 sites. Right: An example of the generation of variants which lost the self-reproductive ability but became capable of reproducing smaller offsprings than themselves (indicated by white circles). This is the snapshot taken in the run with an ancestor of species 15 and a space of 200 × 200 sites.

square root of the size of the space. According to this, it can be expected that the larger the ancestor becomes, the higher the probability of extinction of the self-reproductive loops (i.e., the whole system falling into a static configuration or a short-period repetition of several configurations) becomes, and this is actually observed in Table 6.2. This implies that the continuous self-reproduction of the loop needs enough space in proportion to the size of the loop.

6.2.7 Modifying the initial structure

One explanation for why the evolution toward fitter self-reproductive species did not emerge in the previous experiments may be that the loop in that stage did not have the mature capability of injecting enough genes into its offspring if the form of its arm was altered by some collisions with other structures during the self-reproductive process.

Generally, a genotype in a self-reproducing loop does not describe the whole shape of its phenotype, but only an edge and a corner of the loop. The square formation of the loop emerges as a result of the repeated ontogenetic process of translating such genotype into phenotype. The mechanisms of closing an offspring loop, detaching the offspring from the parent, and germinating a new construction arm, are all dependent on the physical laws (state-transition rules) of the CA world, instead of being described in the genotype. Figure

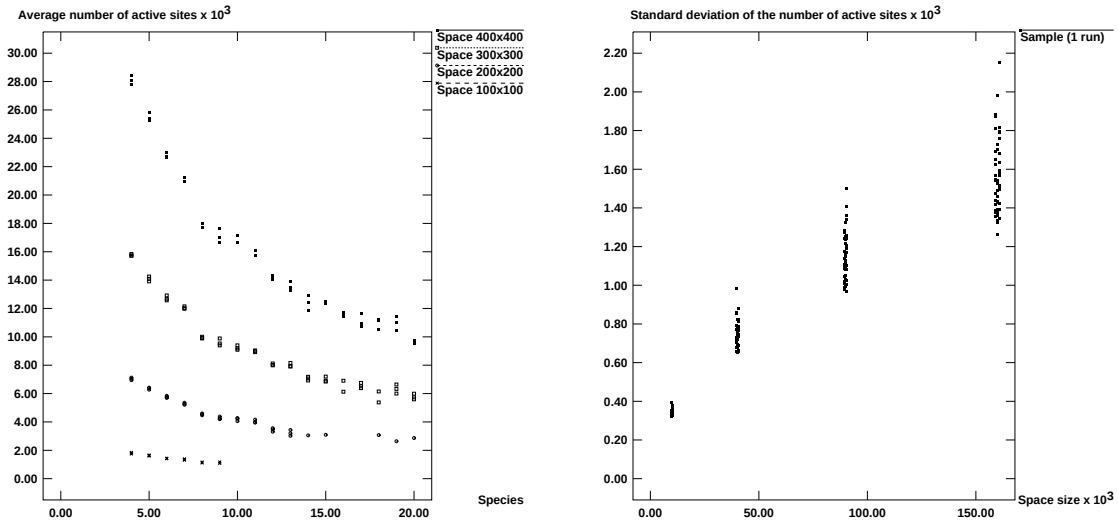


Figure 6.12: Averages of the numbers of active (non-zero) sites contained in the space (left) and standard deviations of their fluctuation (right) calculated according to the results of preliminary experiments. Data from transient periods at the beginning of each run (t=0–9999) were deleted. The cases in which the self-reproductive loops became extinct before t=50000 were also excluded. The horizontal axes represent the species of the ancestor in the left graph, and the size of the space in the right graph, respectively. The average number of active sites in the space decreases approximately in inverse proportion to the size of the species, while the standard deviation of the number of active sites are roughly constant and independent of the species if the size of the space is fixed, and increases approximately in proportion to the square root of the size of the space.

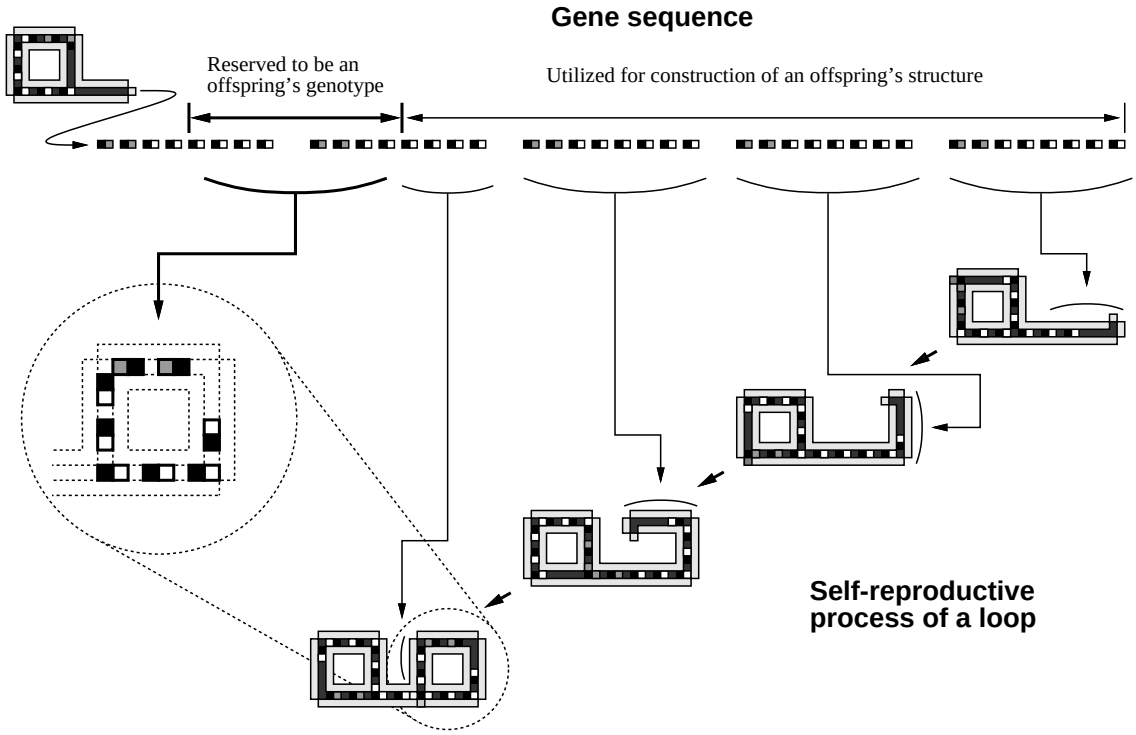
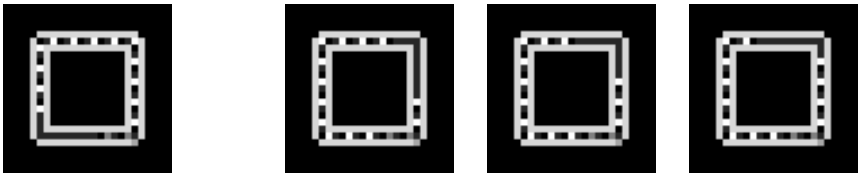


Figure 6.13: Manner of utilization of a gene sequence in the loop. A gene sequence of a little less than four repetitions of the genotype in the loop is utilized for construction of an offspring's structure. Then, a new genotype of the offspring is made of genes contained in the subsequent region of the sequence. These genes can be sufficient, insufficient or even redundant, depending on their order in the sequence, which would affect the self-reproductive ability of the loop.

6.13 shows the manner of utilizing the gene sequence in this process. A gene sequence of a little less than four repetitions of the genotype in the loop is used for the construction of an offspring's structure, and the genes contained in the subsequent region of the sequence is injected to the offspring to be its new genotype. This region can end up with genetic information which can be sufficient, insufficient or even redundant, depending on the order of genes in the sequence. Therefore, it is likely that the order of genes would affect the self-reproductive ability of the loop.

Based on this idea, we looked for new genotypes of the *evoloop* which would have stronger self-reproductive ability than before by examining various genotypical patterns. This effort by trial and error fortunately resulted in discovering that the *evoloops* with slightly modified genotypes shown in Figure 6.14 have much stronger self-reproductive ability. The function of these genotypes are exactly the same as before, while the order of genes only differs. In the new genotypes, gene '4's are located near the front of a gene stream instead of its end. These genotypes seem convenient for the loop to inject more gene



n-evoloop 2-*evoloop* 3-*evoloop* 4-*evoloop*

Figure 6.14: New genotypes of *evoloops* of species 13 which have stronger self-reproductive ability. The right three loops have new genotypes of strong self-reproductive ability in comparison with the original (the leftmost loop). These figures are standardized into the same state just before germinating the arm for clarification of difference between genotypes.

'7's into its offspring than before when some collisions happen to itself. It must be noted that such genotypes were not viable without the new state-transition rules implemented in this chapter.

In distinction from the old loop, these new loops with new genotypes are tentatively called 2-*evoloop*, 3-*evoloop*, and so on, by prefixing the number of gene '7's in front of gene '4's. According to this naming manner, the old loop should be called *n-evoloop*.

6.3 Experiments

6.3.1 Self-reproduction and evolution in finite space

We conducted full-scale experiments of breeding the *evoloops* with new genotypes. The size of the space was decided in consideration of both computing speed and feasibility of visualization, to be 199×199–201×201 sites with periodic boundary conditions. The loops of species 10–13 were selected to be ancestors as they were the largest species which rarely became extinct in the spaces of the above-mentioned sizes. The 2-, 3-, and 4-*evoloops* were all examined.

These experiments resulted in that, surprisingly, in almost all cases the *evoloops* spontaneously varied and evolved through direct interaction of phenotypes, the whole population gradually evolved toward fitter (smaller) species, and finally the space was filled with the fittest species. All the results are shown in Table 6.3.

A result using 2-*evoloops* of species 13 in the space of 200×200 sites is shown here for a typical example. In almost all other cases, the behaviors of the whole system are qualitatively same as this.

Figure 6.15 shows the temporal development of configuration in the above-mentioned case. At first an ancestral loop is set alone in the center of the space. When the simulation begins, the ancestral loop soon proliferates to all the space. Then, self-reproduction and

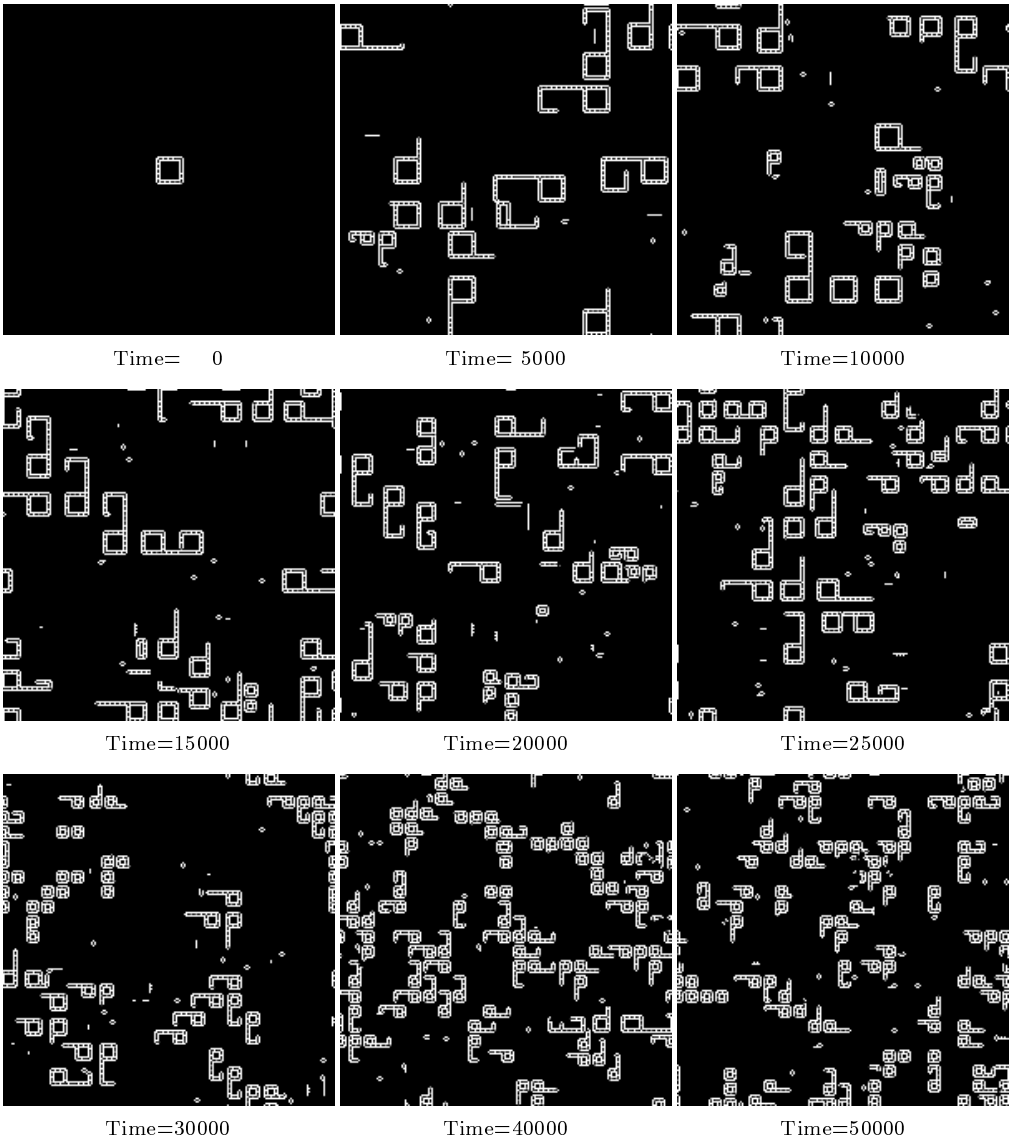
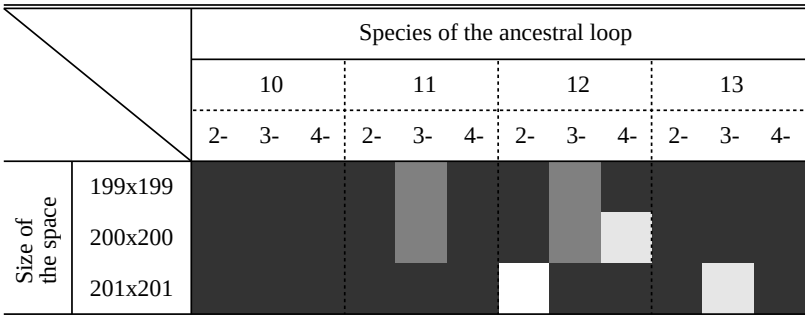


Figure 6.15: Temporal development of configuration in the evolutionary process of 2-*evoloops*. The ancestor is of species 13. The space is of 200 × 200 sites with periodic boundary conditions. As time proceeds, smaller loops emerge and dominate larger ones. The whole population gradually evolves toward smaller species, and finally the space becomes filled with the loops of species 4 which is strongest in this world.

Table 6.3: Results of full-scale experiments of breeding *evoloops* with new genotypes. The species of the ancestral loop ranges from 10 to 13. The 2-, 3-, and 4-*evoloops* were all examined. The size of the space ranges from 199 × 199–201 × 201 sites. Heavy gray indicates the rapid evolution of loops emerged in that case, while dark gray indicates the gentle evolution of loops. Light gray means that the evolution did not occur but the self-reproductive activity of loops continued for 50000 updates. Blank means that the whole system fell into a fixed point or a short-period limit cycle before 50000 updates.



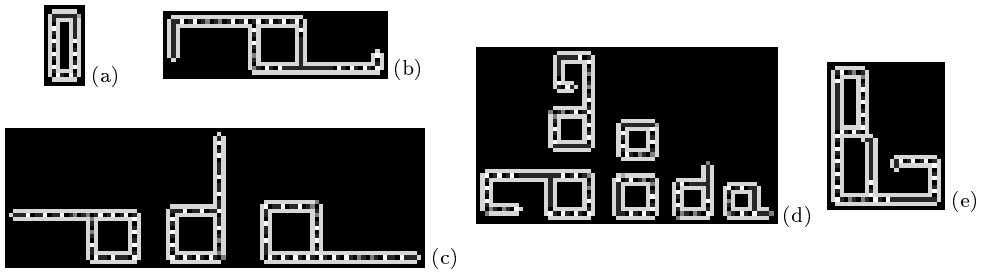


Figure 6.16: Examples of variants produced through the repetition of self-reproduction and structural dissolution of *evoloops*. (a) a loop with no gene '4', (b) a loop having two arms, (c) a loop which does not reproduce itself but does a larger offspring than itself, (d) a loop which does not reproduce itself but does a smaller offspring than itself, and (e) a loop in an indescribable shape.

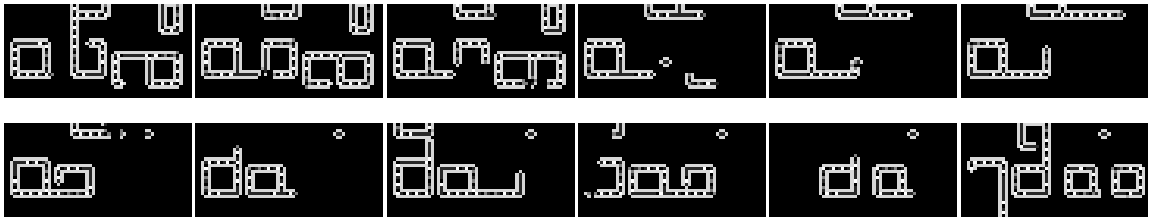


Figure 6.17: Example of variation of *evoloops* occurring through the direct interaction of phenotypes. The length of the arm of a loop of species 9 (the leftmost one) was altered by collision with another loop, which produced a little smaller rectangular variant. This variant generated an offspring of species 7, which fortunately had the capability of self-reproduction.

structural dissolution of loops begin to happen frequently in the space, which produce various kinds of variants such as sterile loops, loops with two arms, loops not self-reproducing but reproducing smaller offsprings than itself, and so forth, as shown in Figure 6.16. A self-reproducing loop of smaller species also emerges by accident from this melee, and once it appears, it is naturally selected to proliferate in the space, due to its quick self-reproductive ability. Such an evolutionary process develops in the space as time proceeds, and eventually, the whole space becomes filled with the loops of species 4 which is the strongest species in this world.

The cause of evolution in this world is chiefly the direct interaction of phenotypes such as the collision of two loops and the crash of a loop into a debris structure, which may change the length of their construction arms. A typical example of such phenomena is shown in Figure 6.17. It is quite characteristic of this evolutionary process that the variation in this world occurs first on phenotype (not on genotype) of the offspring being produced and then it consequently leads to alteration of genotype. This manner of variation is opposed to the idea of mutation we usually consider.

Figure 6.18 shows the temporal development of numbers of living loops and their genealogy in the above case. It is clearly observed in these graphs that various species of *evoloops* are produced in the course of evolution, and species 4 finally exterminates the other species. The right genealogy indicates that the variation occurring in this world has some tendency to happen toward smaller species, but also happening toward larger ones in relatively low probability. Anyway, the whole system seems to evolve toward the smallest species 4 approximately in proportion to the elapsed time. In addition, it is found in this genealogy that the larger species sometimes exterminates the emergent smaller one which should theoretically have higher fitness than the larger one has. This indicates that the selection in the evolution of life may be affected to some extent by local, unpredictable[2]

[2]Here what we mean by "unpredictable" is that it cannot *a priori* be predicted before a simulation is carried out.

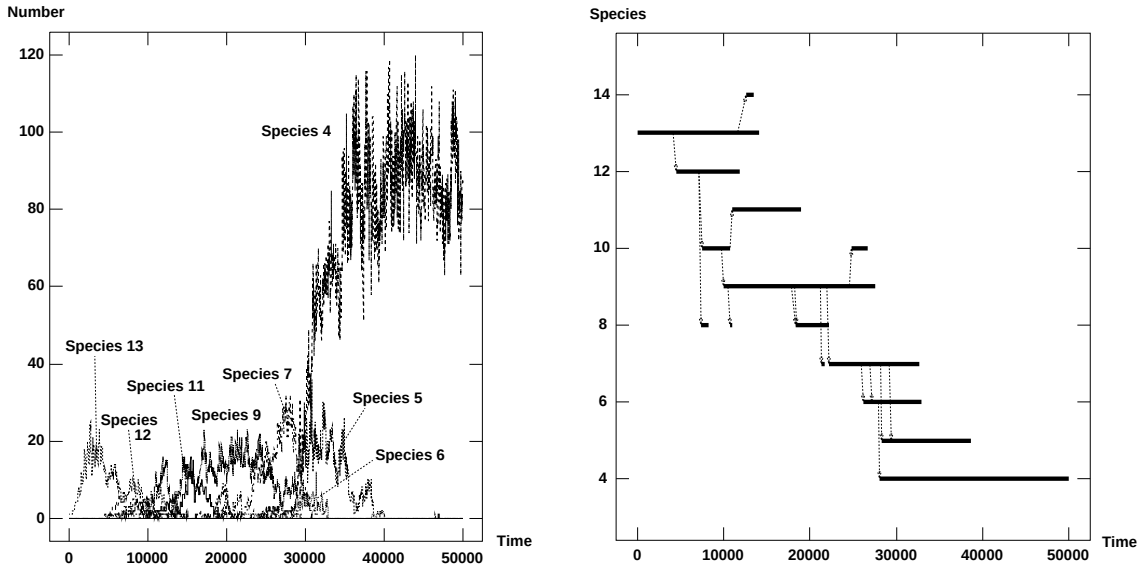


Figure 6.18: Temporal development of numbers of living *evoloops* (left) and their genealogy (right) in the case of Figure 6.15. It is observed that various species of *evoloops* are produced in the course of evolution, and species 4 finally exterminates the other species. The whole system seems to evolve toward the smallest species 4 approximately in proportion to the elapsed time.

conditions as well as the difference of fitness values of competitive species.

6.3.2 Relation between resource size and evolvability

In the following several sections, we shall investigate the evolvability of the *evoloop* from a viewpoint anticipating the future implementation on hardware.

We expect that the size of the artificial world being simulated can be enlarged greatly by implementing it on hardware. The basic reason why we hope this is that we believe that the larger the scale of simulation becomes, the more the complexity and diversity of the artificial system evolves. Thus, we first examine this assertion within the *evoloop* world.

In order to investigate the relation between the size of the available resources and the evolvability of loops, the 2-*evoloop* of species 13 was bred in the spaces of various sizes ranging from 200×200 to 2000×2000 sites. Figure 6.19 indicates the temporal development of the diversity (the number of species living in the space) of *evoloops* in several cases with different spatial sizes. Since the number of possible species of the *evoloop* is limited, the diversity is saturated around the case of 1200×1200 sites, but nonetheless, it is possible to say with respect to relatively small spaces before the saturation that, the larger the size of space becomes, the more easily various species can appear and coexist with each other (for a while) due to larger capacity of space. Figure 6.20 is an example of the configuration

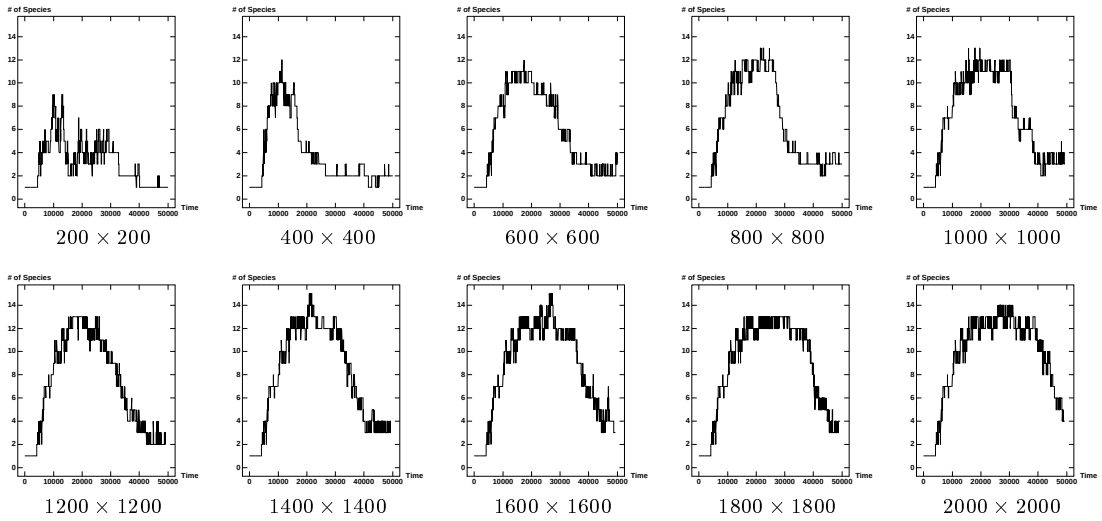


Figure 6.19: Temporal development of the diversity (the number of species) of *evoloops* in several cases with different spatial sizes. The ancestor is a 2-*evoloop* of species 13 in all cases. Figures placed under each graph represent the size of space in that case. It is observed with respect to the cases with spaces smaller than 1200 × 1200 that, the larger the size of space becomes, the more easily various species can appear and coexist with each other due to larger capacity of space.

in such a state of diverse evolution emerging in a large space.

This result supports the above-mentioned assertion that large resources affords large diversity of evolution, and also suggests that, when we design more complex organisms (like *Tierran* creatures shown in Fig. 4.1) on CA in the future, we will need much larger resource in order to keep their diversity.

6.3.3 Fault tolerance

Fault tolerance is an important feature of the model when its possibility to be implemented on hardware is considered. The more tolerant of faults the model becomes, the lower the level required for the hardware performance becomes, accordingly the easier the enhancement of circuit integration becomes. Note that the word "fault tolerance" in this dissertation means the endurance of the organisms on CA against the bad workings of the CA world *itself*, being different from the above-mentioned adaptability of the organisms which is defined and measured on the assumption that the workings of the CA are correct.

Here is the fault tolerance of the *evoloop* measured. Two kinds of faults are supposed in this measurement: The errors of state-transition rules which simulate unstableness of hardware modules, and the distribution of state-fixed blocks in the space which simulates damage of hardware modules. The probabilities that the self-reproductive activity of *evoloops* will continue for 10000 updates are measured with the rates of the above two

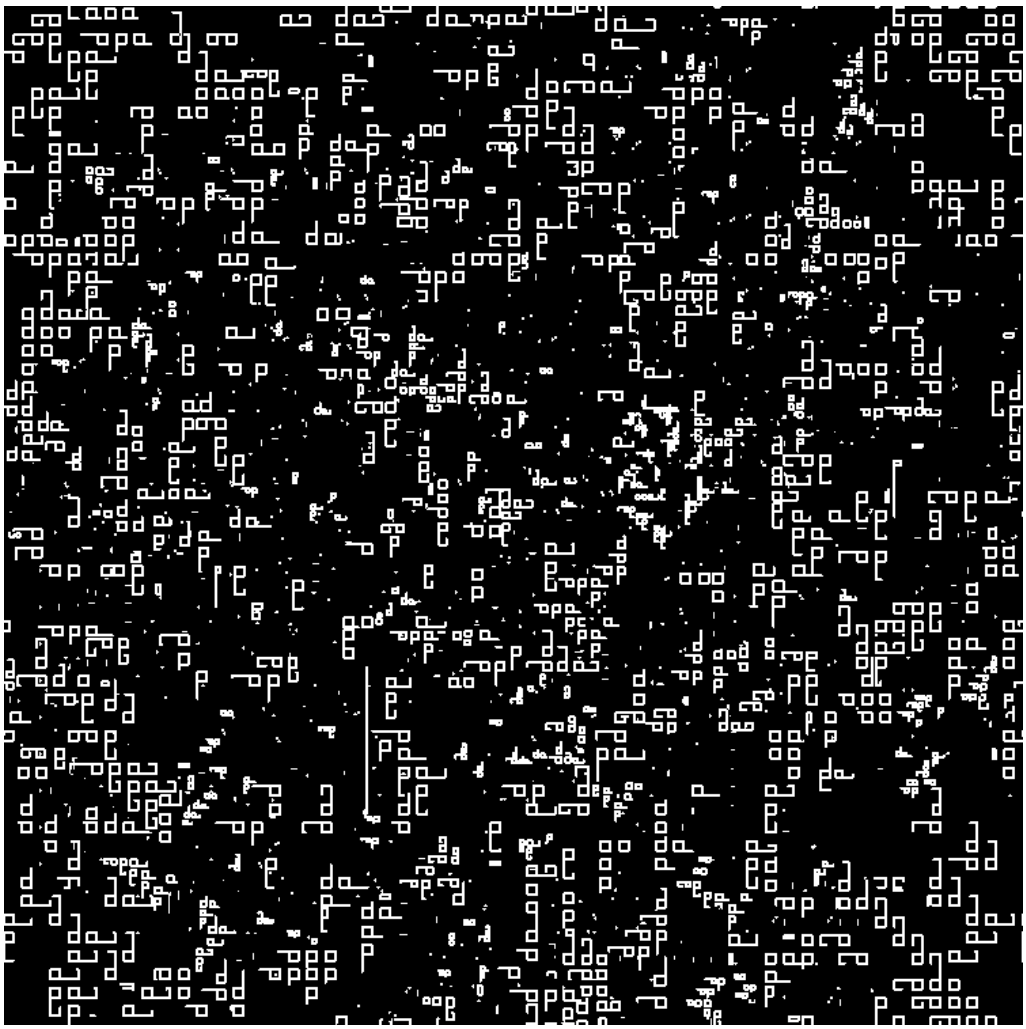


Figure 6.20: Snapshot taken in the evolutionary process of *evoloops* in the space of 1500 × 1500 sites (Time=20000). The ancestor is a 2-*evoloop* of species 13. The space is islanded with the colonies of loops of various species.

Table 6.4: Probabilities that the self-reproductive activity of *evoloops* will continue for 10000 updates with two kinds of faults provided. The probabilities in each case were calculated based on the results of ten simulations of breeding 2-*evoloops* of species 10 in the space of 200 × 200 sites with different seed values of random numbers. Emphasis is added on the probabilities of 0.5 and over.

		Rate of errors of rules ($\times 10^{-6}$)				
		0.0	2.0	4.0	6.0	8.0
Rate	0.0	**1.0**	**0.8**	**0.8**	0.3	0.0
of	0.5	**1.0**	**0.5**	0.3	0.1	0.0
fixed	1.0	**0.7**	0.3	0.1	0.0	0.0
blocks	1.5	0.2	0.2	0.1	0.0	0.0
($\times 10^{-3}$)	2.0	0.1	0.1	0.0	0.0	0.0

faults being changed. The results are indicated in Table 6.4. It is understood from this table that the *evoloop* is actually tolerant of faults to some extent. The main reason for this is that the *evoloop* can eliminate irregular structures caused by faults quickly from the space, owing to its capability of structural dissolution, which would be regarded as quite a favorable property for hardware implementation. It is expected that the *evoloop* will be more tolerant of faults with further modification of the state-transition rules to a simpler and more robust form.

6.3.4 Emergence of self-reproducing organisms

We can observe the emergence of self-reproducing organisms from empty space by introducing a few additional state-transition rules and embedding small devices into the space. The additional rules are

$$\frac{\texttt{CTRBL->I}}{\texttt{00011->2}} \quad , \quad \frac{\texttt{CTRBL->I}}{\texttt{10000->1}} \quad , \text{ and } \quad \frac{\texttt{CTRBL->I}}{\texttt{200**->2}} \quad (* = 1, 4, 7) \, ,$$

which are for the situations occurring near the devices introduced below.

The device used here is a non-deterministic finite-state automaton as small as a single site of the CA, and behaves stochastically to generate possible gene sequences meaningful in the *evoloop* world. Figure 6.21 depicts the state-transition diagram of this device. It generates random gene sequences filled with a mixture of a gene '7' and a pair of gene '4's, which cause the growth of pre-biotic arms in the space. Self-reproducing organisms can emerge by accident from the interaction of these arms. Figure 6.22 shows a typical example of the emergence of self-reproducing organisms realized by using this device.

This phenomenon may be applicable for searching for possible structures of viable organisms in an artificial world where only a grammar of their gene sequences is known.

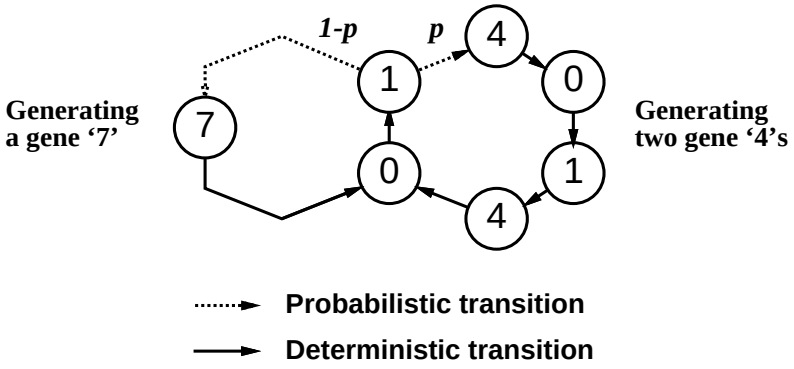


Figure 6.21: State-transition diagram of the stochastic device embedded in the space. It can generate random gene sequences filled with a mixture of a gene '7' and a pair of gene '4's, which can be translated to structures correctly in the *evoloop* world. Probability p is set to be 0.05 in this study.

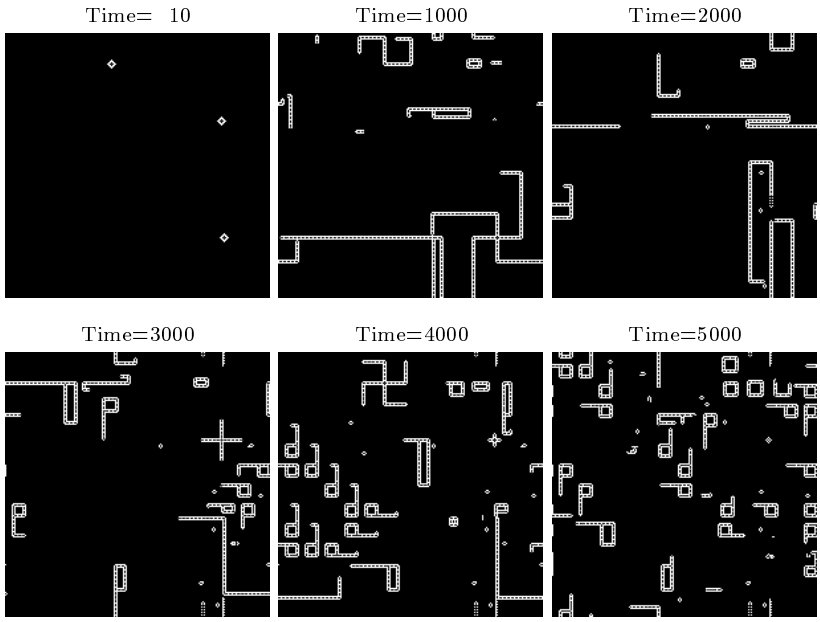


Figure 6.22: Example of the emergence of self-reproducing organisms from empty space. Three stochastic devices are randomly distributed in the space of 200×200 sites. These devices generate random gene sequences which cause the growth of pre-biotic arms. Self-reproducing organisms can emerge by accident from the interaction of these arms, and gradually proliferate over the space.

This kind of method to embed small additional devices in a uniform array of machine modules would be easier and more practical when they were implemented on the hardware.

6.4 Problems

Though the *evoloops* showed the interesting behavior of spontaneous evolution which had not been reported before, we could not observe in their world either punctuated equilibrium of evolution or symbiosis of different species, which had been reported in the other evolutionary systems[37, 41]. The main reason of this is that the *evoloops* have no ability to interact with each other in a functional way so that they cannot build complex relations by altering mutual fitness landscapes. In other words, the fitness landscape of *evoloops* is fixed throughout the run, where they merely adapt to a physical environment—the static space comprised of a fixed number of finite sites. Some plans for solutions of this problem are introduced in section 8.2.

In addition, we should note here again the difficulty of observing the behavior of evolutionary systems in CA space. Since the meaningful structures in CA are merely spatial collections of local states, all the judgments—which structure is living or dead, how the genealogy of organisms is formed, when and where the variation occurs, e.g.—are left to the observer. This means that the observation of such a world needs enormous computation for image processing in order to extract meaningful information from the temporal development of configuration of the whole CA space. With respect to this problem, we are challenging to solve it, as mentioned before, by utilizing another CA specialized for image processing, since the image processing itself is a good target for the application of CA.

Chapter 7

DISCUSSIONS

In this chapter, several issues with respect to the features of the SDSR loops and the *evoloops* are discussed from both biological and engineering viewpoints.

7.1 Significance of structural dissolution

Almost all models of artificial life on CA so far proposed have shown no more than regular behaviors that their designers have intended to embody, because those studies have focused on merely the artificial realization of regulated phenomenon such as self-reproduction. As opposed to them, in this study we succeeded in transforming the regular but abiotic behavior of Langton's SR loop into a more dynamic, lifelike form by introducing a form of death into it through a new dissolving state '8' being added to the set of states of the CA.

Then, at first, we would like to discuss the significance of the structural dissolution for the system, by considering the behavioral difference between the SR and SDSR loop. Both loops are identical to each other except for the rules concerned with dissolving state '8'. As a result, SDSR loops show several unique phenomena never observed in the world of SR loops, such as

(1) an ability to overcome external difficulties to some extent by means of partial structural dissolution,

(2) a significant increase in number of living loops when bred in infinite space,

(3) a continuous self-reproduction in finite space,

(4) a production of many merged loops caused by direct interaction of phenotypes in the course of self-reproduction, and

(5) a competitive exclusion acting between loops of different species living in the same finite space.

Among them, (2) is considered to be substantially the same as (3) since the loop gained the possibility of continuous self-reproduction in the inner area of the colony. Similarly, (4) is considered to be a side effect of (3) since it was caused by the fact that rare events to produce merged loops happened occasionally in the long continuation of self-reproduction. Moreover, (5) is regarded as a phenomenon resulting from both (3) and (1) because they gave rise to (5) by the continuation of self-reproductive activities of competing two species and also by the ability of loops to overcome external difficulties (the opponent species in this case).

According to the above consideration, the substantial meaning of structural dissolution lies in (1) and (3). The difference between them is the level at which structural dissolution occurs. In (1), the structural dissolution occurs to a partial structure of the loop, while in (3) it occurs to the whole structure of the loop. Though we usually view the latter as the cessation of life, the level of dissolution is actually not essential to the consideration of the behavior of the whole system. We may derive the significance of structural dissolution from the feature common to (1) and (3); namely, in these two phenomena, structural dissolution functions as a kind of negative feedback mechanism which dissolves a subsystem becoming inconsistent with the environment. Due to this mechanism, resources (space in this case) having been owned by the dissolved subsystem will be re-utilized later by the other subsystem, and the repetition of such processes provides the whole system with dynamic, adaptive stability, and furthermore, probability of evolution. For instance, an individual can secure its living state by dissolution of partial organs, and a colony can continue its dynamic behavior by dissolution of individuals.

It should be recalled here that the introduction of structural dissolution to CA conducted in this study is easily applicable to other general CA too, because, as mentioned in section 4.2, the introducing method can be summarized into a very simple procedure as follows:

- To add a dissolving state (which turns into a quiescent state at any time) to the set of states of the CA,
- to define rules concerning the contagion of the dissolving state and the dissolution of its adjacent structures, and
- to let all undefined situations turn into the dissolving state.

By this method, most CA would become able to dissolve unfit structures from the space by generation and propagation of the dissolving state, which would lead to more dynamic, adaptive behavior of the CA than before.

7.2 Strategic death based on structural dissolution

Next, we shall give some consideration to the relation between death and structural dissolution introduced in this study.

Note again here that the mechanism of structural dissolution in this study is not described in the genotype of the loops, but rather is embedded in the state-transition rules which govern the dynamics of the CA. In contrast, many genes concerned directly with aging and death have recently been discovered and discussed in real molecular genetics. These discoveries suggest that death is effectively preprogrammed in real creatures by genetic means. From these facts it is easily understood that we should not make the jump to identify the structural dissolution introduced in this study with death in real life.

A question about death may rise before us here. If the mechanism of death were genetically programmed in individuals, those individuals should be destined to be driven into extinction in the course of natural selection by the emergence of immortal individuals which had lost their programmed death, as was pointed out by Todd[54]. Nevertheless, genetic programs implying aging and death actually exist in real creatures while avoiding being eliminated through natural selection. How, then, should we interpret these contradictory facts?

We suppose that the notion of structural dissolution modeled in this study may be helpful in giving an answer to this question. The hypothesis we offer here is that death programmed by genetic means may be effective and remain in the population if a primitive mechanism of structural dissolution is embedded in the basic laws of Nature, such as the state-transition rules of CA adopted in this study. In other words, if the world is furnished with the rule that everything will change sometime no matter how much you invest to preserve it, the strategy of dynamic development of structures with positive dissolution of them may be more efficient than the strategy of static maintenance of fixed structures, and as a consequence, death programmed in genes may be selected by living organisms.

The other extreme of self-preservation is to force continuation of existence by constructing extra-strong structures in a static equilibrated system without dissolution, such as the structure of carbon atoms in diamond. However, such structures are generally simple, regular and static, and cannot be the source of dynamic behavior characteristic of life. As life is a structure which emerges in a system far from equilibrium, there may always be an essential probability of the collapse of its structure. In such a world equipped with a fundamental possibility of structural dissolution, it would be likely that strategic death has an advantage over other strategies and becomes the substantial characteristic of life.

With respect to candidates for the primitive mechanism of structural dissolution in real life, we can enumerate a number of facts with the second law of thermodynamics at the head of the list, such as that biochemical macromolecules contain a lot of non-covalent

bonds relatively easy to break, that biochemical macromolecules are always suffering perturbation given by electric polarity of H_2O molecules surrounding them, and so forth. Furthermore, it is also likely that these facts interact with one another to produce more complex effects of a higher order. Compared with such complex real phenomena, it is obvious that the structural dissolution introduced to the CA in this study is quite an abstract phenomenon realized by a simple mechanism of propagation of dissolving states. Therefore, at this stage, we should regard structural dissolution in this study as nothing more than a mathematical model of a qualitative aspect of structural dissolution in real life, which does not copy any specific phenomenon actually occurring in the real world.

7.3 Evolution driven by direct interaction of phenotypes

The results obtained in this work can be regarded as a unique example of evolution in which the variation occurring on phenotypes by their direct interaction leads to the variation of genotypes. In this sense, it may have a little resemblance to Lamarckian evolution.

On the other hand, we usually think that the phenotype of life develops mostly according to its genotype, so that the evolution of life is caused by probabilistic change which might occur to its genotype. This notion is based on the idea that a biological cell is regarded as a universal constructor which can control external/internal environments and faithfully construct another cell according to DNA "tapes". Of course, this is almost certainly true with respect to sophisticated life forms such as eukaryotic organisms including human beings.

However, as noted by Langton[27], there can be little doubt that life at the ancient dawn was *not* a universal constructor, because the universal constructor must generally be made of an enormous number of components[3, 11, 39, 57, 58], which the ancient life was unlikely to possess. At that time, the genotypes of living objects may have been physical structures at the same scale as their phenotypes, so that the genotype may have had some phenotypical character, and *vice versa*. Thus, it is very likely that the evolution of life at that time was accomplished not only by probabilistic change of genotypes but also by such environmental factors as direct interaction of phenotypes—in other words, such environmental factors could alter how the genotype was interpreted into a phenotype.

Since Langton's aim in inventing the SR loop was to create a model of primitive life forms in extremely small size which did not have the universality of construction/computation but was capable of self-reproduction, genotype and phenotype of the SR loop were both spatial structures of the same scale and had a potential to influence each other. This feature has been inherited by the SDSR loop and the *evoloop*. In the *evoloop* world, it actually happened that the phenotypical interactions such as the collision of two loops altered the genotypical information, which consequently drove their evolution.

Such an evolutionary process emerging in the *evoloop* world may bear a close resemblance to the beginning of evolution of primitive life of small complexity. In such a world, the individual organism must have evolved by not only genetic mutation but also interaction with the external environment including other individuals.

7.4 Characteristics as artificial life implemented on CA

The *evoloop* contrived in this dissertation has several characteristics as a model of artificial life implemented on CA.

The most important feature is that the *evoloop* is the first to realize the spontaneous evolution of self-replicators through variation and natural selection in the CA space. With respect to the evolution of self-replicators in a CA space, Chou et al.'s model[9] (shown in Fig. 3.12) seems most successful among the former artificial life on CA. However, their principal objective was to observe the emergence of self-replicators out of the "primordial soup" rather than to realize their evolution, thus the evolutionary behavior obtained in their study was no more than a repetition of a process where the smallest loops emerged at first, they were forcedly made to mutate toward the larger ones by the external mutator states distributed over the space, and eventually they became extinct as they evolved to be too large.

On the other hand, the main focus of our study is on the evolutionary process itself. In this study, the initial configuration was an empty space but *a priori* populated by a sole ancestral loop which had the self-reproductive ability, and no devices for explicit promotion of the evolution of loops was provided. Therefore, the evolutionary process emerged in the *evoloop* world is a truly spontaneous one produced by the variation and natural selection acting on the population of *evoloops*.

In addition, we can give it as another characteristic of the *evoloop* that the CA space used in this study has an extremely small complexity compared to the other models. For example, Chou et al.'s model mentioned above was 256-state CA with a Moore neighborhood, thus the size of the state-transition rules is $256^9 = 2^{72} \approx 5 \times 10^{21}$ when expanded into a form of a look-up table, while our CA is 9-state with a von Neumann neighborhood, thus the size of the rules is $9^5 = 59049 \approx 6 \times 10^4$. This is probably the smallest complexity at the present time among the CA models proposed for simulating the evolution of self-replicators.

It would be worth doing here to extract the know-how of constructing state-transition rules from the experience of implementation of the *evoloop*, because it is generally rather difficult to construct a system which behaves in such an intriguing way as the *evoloop* does by utilizing a CA space of small complexity like that used for the *evoloop*. We can give several important techniques employed in constructing the state-transition rules of

the *evoloop*, as follows:

- To classify the states of CA into several categories and decide their respective roles roughly before the construction of state-transition rules. This will prevent unnecessary confusion from happening in the constructing task of the rules. For example, the states used in the *evoloop* implemented in this study are classified into three categories—fundamental elements ('0', '1', '2'), signals ('3', '4', '5', '6', '7') and a dissolving state ('8') (see Table 6.1). The signals can be classified further into genes ('4', '7') and other signals ('3', '5', '6').

- To divide the behavior of the virtual state machines into several phases and prepare a clear plan for assignment of functions to states in each phase. This will enable us to make the most efficient use of a limited number of states. As for the *evoloop*, state '3' and '4' are assigned with different functions in different phases as shown in Table 7.1. The state '3' is in particular fully utilized in the CA of the *evoloop*, which we could attain by making the definite plan for its functions beforehand.

- To let the information processing in the inner structure of the virtual state machines be explicitly driven by the arrival of some propagating signal rather than by the emergence of some specific situation; namely, the processing mechanism should be able to wait for the arrival of that signal. This will make the situation in which the mechanism works the same form, which will facilitate the enhancement of the adaptability of that mechanism. For example, in the phase of bonding of the tip and the root of the arm of the *evoloop*, the bonder '3' is designed to wait for the arrival of a state '7' propagating from the parent (see Fig. 6.3). This operation transforms the locations of signals in the vicinity of the bonding point (possibly different as the case may be) into the same form.

It is certain that due to these techniques we could construct the state-transition rules of the *evoloop* more efficiently than we had expected.

The breakdown of the state-transition rules of the *evoloop* is carried in appendix D for reference, in which the principal part of its state-transition rules is classified into six phases and other general rules, and moreover, every rule included in six phases is accompanied with a brief comment about its function. To read this breakdown carefully may be useful for the readers to comprehend the know-how of constructing state-transition rules introduced in this section.

7.5 Characteristics as an artificial evolutionary system

Next, the characteristics of the *evoloop* as an artificial evolutionary system are evaluated here through the comparison with other evolutionary systems invented thus far.

Table 7.1: Functions of state '3' and '4' in different phases of the self-reproductive process of the *evoloop*.

Phase	State	
	'3'	'4'
To turn the tip of the arm left	Left indicator	Gene
To bond the tip and the root of the arm together	Bonder	—
To germinate a new sprout of the arm	Sprout generator	Gene
To lengthen the new sprout of the arm	Sprout capper Sprout finisher	Sprout guide

One of the most characteristic points of the *evoloop* is that it has no central operating system to maintain information about the activities of living individuals. All of the other evolutionary systems so far constructed[1, 37, 41] needed rather complicated operating systems for identifying living or dead individuals, simulating their activities, inducing mutations to them, and so on. On the other hand, as for the *evoloop*, though it is provided with only simple physical laws which govern the micro-level behavior of the universe, it can produce an evolutionary process where the genotype and phenotype of organisms definitely exist in the space and the individuals vary and compete with each other through the interaction among them, thus the evolutionary process in the *evoloop* world would be considered as a completely emergent, self-organized one.

The other characteristic of the *evoloop* is that it does not involve any stochastic operation at all. There are no random flip of genetic information in the *evoloop* world, where all the gene '4's and '7's emerge by only transcription of them at the T-junction of the pathways, and disappear by only either translation at the tip of the arm or structural dissolution. This indicates that the evolution of life can result from completely deterministic processes.

This feature may be analogous with the case that the CA model of Turing patterns[56] (patterns emerging on the surface of bodies of animals/plants produced by the complex behavior of reaction-diffusion chemical systems) could spontaneously generate irregular patterns such as branching or breaking of zebra stripes by utilizing information implicitly lying in the initial configuration[62], while the differential-equation-based models for them could not generate such irregularity without operations being added explicitly from the outside[34]. The important point common to the *evoloop* and the CA model of Turing patterns is that they are able to make use of disordered information hiding in low-level substructures as the seeds of unpredictable behavior emerging at the high-level superstructures, because in these models the subject phenomena are built from the most low-level

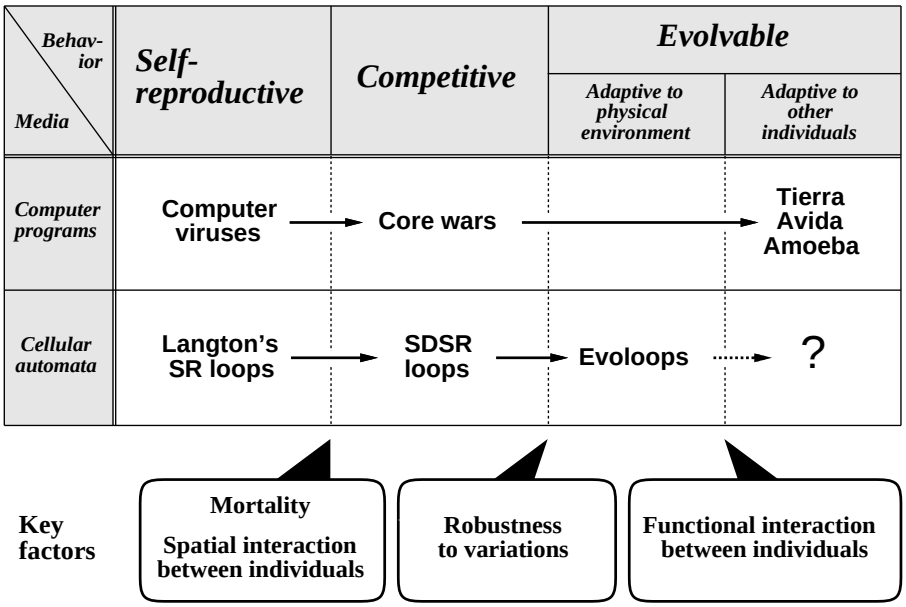


Figure 7.1: Analogy between the development of digital organisms made by computer programs and the development of self-reproducing loops on CA. The key factors common to both kinds of artificial systems in enhancing their behaviors are also shown at the bottom.

elements in a bottom-up way, instead of being described directly in high-level formulae. This would be the main reason why the *evoloop* does not need any stochastic operation.

7.6 Evolvability of artificial evolutionary systems

We may derive from this study some insights on the evolvability of artificial evolutionary systems. Figure 7.1 shows a rough analogy between the development of digital organisms made by computer programs[1, 12, 13, 37, 41] and the development of self-reproducing loops on CA including the SDSR loop and the *evoloop*. The behaviors of these artificial systems are classified here into three categories: Self-reproductive, competitive, and evolvable; the last category is further divided into two: Adaptive to physical environment and adaptive to other individuals.

It would be possible to extract the factors common to both kinds of artificial systems in the same class of this analogy. The key factors to create competitive systems is obviously both the mortality of the individuals and the spatial interaction between them. However, they are not sufficient in advancing artificial systems to the evolvable class. The most important factor common to evolvable systems would be the robustness of organisms to variations. For example, the famous evolutionary system *Tierra*[41] mentioned in chapter 4 met with success by making both its instruction set and its addressing mode quite robust

to genetic operation such as mutation and recombination of program codes. This factor also forms the main difference between the SDSR loop and the *evoloop*.

Of course, the *evoloop* is presently not in the same class as *Tierra*, as it adapts only to the physical environment because a fourth key factor is lacking, i.e., functional interaction between individuals which causes the emergence of complexity and diversity of digital organisms in *Tierra*. However, we expect it to be possible to introduce this feature into the *evoloop* in the near future, which will be mentioned in the next chapter.

If such a complex evolutionary system like *Tierra* could be constructed on a simple deterministic CA space as a successor to the *evoloop*, it would be expected to have some similar characteristics inherited from the *evoloop* as follows:

1. The evolution of life in such a system would be realized without any stochastic operations such as random mutation of genotype,

2. there would be no need of a central operating system to maintain/evaluate information about the activities of all living individuals, since any particular information about the individual organism is maintained on the configuration of the CA space where it resides, and

3. such a system would be intrinsically very suitable for massively parallel processing, since all the behaviors of the system emerge from only local computations between neighboring sites.

These imply that, in the near future, we would be able to create extraordinary large-scale evolutionary systems on a fine-grained superparallel machine environment by using extremely simple algorithms, which would greatly advance our knowledge of both natural and artificial life. Such systems might also contribute well to the progress of the efforts to implement novel systems of parallel computation by using evolutionary CA[10, 18, 31, 50] which are energetically studied these days.

Chapter 8

CONCLUSION AND FUTURE EXTENSIONS

> *... It is clear that the invention of useful models that require such machines has been and will continue to be an important stimulus to their evolution. Conversely, the prospect of such dramatic increases in simulation speed and size strongly encourages the further development of models and modeling techniques which exploit the strengths of these machines.*
>
> — *T. Toffoli and N. Margolus, Cellular Automata Machines* [55]

8.1 Summary of this study

In this study, we discussed whether it is possible to construct evolutionary systems on a simple deterministic CA space in which artificial organisms spontaneously evolve toward fitter forms through variation and natural selection. The answer we give to this problem in this dissertation is in the affirmative, as we showed an instance of such evolutionary systems, the *evoloop*, constructed after Langton's SR loop. This system was implemented by mainly three steps—first, a new dissolving state was introduced into the set of states of its CA in order to embody death and natural selection of the loops; second, its state-transition rules were fully rewritten in order for them to become more tolerant to the fluctuation of environmental conditions; and finally, its initial structure was slightly modified in order to enhance their ability to catch the seeds of variation. The *evoloop* displays quite intriguing evolutionary behaviors, where the loops spontaneously evolve toward the fitter species (i.e., smaller species) through direct interaction of their phenotypes, even under the condition with no stochastic mutation given.

The result obtained in this study suggests that it is possible to create evolutionary systems on a CA space which is an extremely simple medium without any complicated

mechanisms such as explicit procedures for the management of self-reproduction and mutation of individuals living in that world. From the implementation of the *evoloop*, it would also be derived that the most important factors in constructing artificial evolutionary systems are to introduce the mortality of artificial organisms, their interaction, and their robustness to variations into the model. From a biological viewpoint, this work can be considered to present us a special case of evolution where the variation of phenotypes caused by their direct interaction leads to the variation of genotypes. Such evolution might have occurred actually to the ancient life forms of small complexity.

Of course, the *evoloop* in this stage still lacks some important features for diversification of evolution such as the ability to interact with other organisms in a functional way. However, since it is implemented on a very simple deterministic CA space which has only nine states and a five-site neighborhood template, it has considerable room for further modification of the model, and also has considerable potential to be implemented easily on hardware modules in the near future, which will provide us with the fundamental techniques to create extraordinary large-scale artificial evolutionary systems on a superparallel machine environment. We believe that the simulations of the evolution of life in such large scales will produce a great amount of knowledge on the evolution of *life-as-it-could-be*.

At last, we would like to mention that some kind of revolution of hardware technology has been under way these days. For example, the rise of the field programmable gate array (FPGA) chip, which is capable of reconfiguration of its inner circuits, has drastically changed the distinction between software and hardware, and opened up a new field of evolvable machines. It seems very important also for this study to adopt such new technology to produce effective implementation of evolutionary systems.

8.2 Future extensions

As the *evoloop* introduced in this dissertation is still quite a simple model, we can consider a number of modifications to it. Some of the future extensions presently planned are briefly introduced here.

8.2.1 Facilitating the improvement of CA

The present state-transition rules of the *evoloop* are described by merely a look-up table and several additional operations as shown in appendix C, which is not so convenient in adding further modification or implementing them on hardware in the future. If the rules could be described in a high-level (or at least middle-level) language like CAM Forth language[55], they would become easier and more practical in receiving maintenance, and also would become much easier to be realized in small logic circuits on hardware.

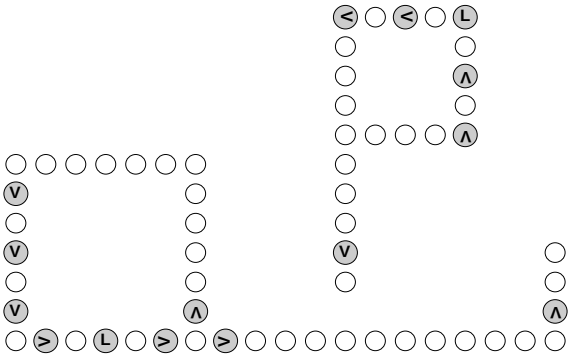


Figure 8.1: Image of unsheathed *evoloops* on CA with weak rotational symmetry. The formation of the loop and the signals in it would become much simpler than now, which would be favorable in translating the state-transition rules into a high-level language.

In order to make it possible to describe the state-transition rules in a high-level language, some devices should be conducted to improve the rules into much more readable and tractable style. There are several known methods to enhance readability and tractability of state-transition rules of CA, such as

- unsheathing the loops,
- introducing weak rotational symmetry to the CA,
- adopting a Moore neighborhood,
- adopting partitioned (multi-data-field) CA in which the information included in a state value is divided into some parts and treated separately[9, 10, 18, 32], etc.

These methods are considered to be effective for the *evoloop* too, and we are planning to apply some of them to the CA before conducting the following extensions. When the above methods would be applied to the CA, the formation of the loop and the signals in it would become much simpler, as shown in Figure 8.1.

8.2.2 Removing restrictions on the shape of organisms

One of the reasons why the evolutionary process emerging in the *evoloop* world is not so complex is that the possible shape of viable organisms in that world are restricted to squares. To prepare new genes for right turning, branching, breaking of the arm, etc., would be effective to break through this restriction and produce more complex shapes than now. Such diversification of the formation of organisms must be necessary for realization of the diverse evolution of interrelation among organisms, thus we are going to introduce these additional genes to the CA next.

However, if we introduce such various genes to the organisms, we also need some additional devices to protect their self-reproductive ability somehow or other, because they no longer form square loops which was crucial for the mechanism of self-reproduction of the *evoloop*. For this purpose, we are planning to adopt the shape-encoding mechanism proposed by Morita et al.[32, 33] which enables any structure to reproduce itself (see also Fig. 3.7). With this mechanism, the organisms (worms as well as loops) continuously encode their own shape into their genotype, which seems in good harmony with the evolution driven by the direct interaction of phenotypes obtained in this dissertation. We expect that the evolution of diversity of artificial organisms would be much accelerated in the *evoloop* world with the shape-encoding mechanism.

8.2.3 Diversifying the function of and the relation between organisms

The most important point for the future extension is to introduce functional interaction between organisms, as remarked in section 7.6, which we are aiming at as a final goal. Functional interaction is necessary for realizing the open-ended evolution and for generating a complex and diverse ecosystem, but there are only spatial interactions in the present *evoloop* world, such as collisions between two loops.

In order to realize the functional interaction between organisms, we anticipate preparing another "memory" CA space containing additional information, and superimpose it onto the identical space by using multi-data-field CA. At the same time, we expect that the organisms in this new CA space should be equipped with the ability to read/write information in the memory CA; namely, the organisms should be granted some ability of information processing or computation. If we could conduct this improvement successfully, the organisms would become able to interact with each other through the access to the same memory CA. The image of this CA model is depicted in Figure 8.2.

This idea of superimposed "memory" CA has much potential to widen the range of subject phenomena to be simulated by using this kind of CA. For example, it may be possible to simulate the competition among creatures concerned with "food", i.e., the materials necessary for constructing and maintaining their bodies, if an additional CA space which contains the information about the amount of finite food is superimposed onto the main CA, besides the activity of capturing such food introduced to the creatures. Moreover, if such layers of CA are designed to interact in some hierarchical way, a phenomenon similar to the food chain may be realized in CA too. These modifications of the model would generate unpredictable evolution of artificial organisms and even the formation of their society in a CA space.

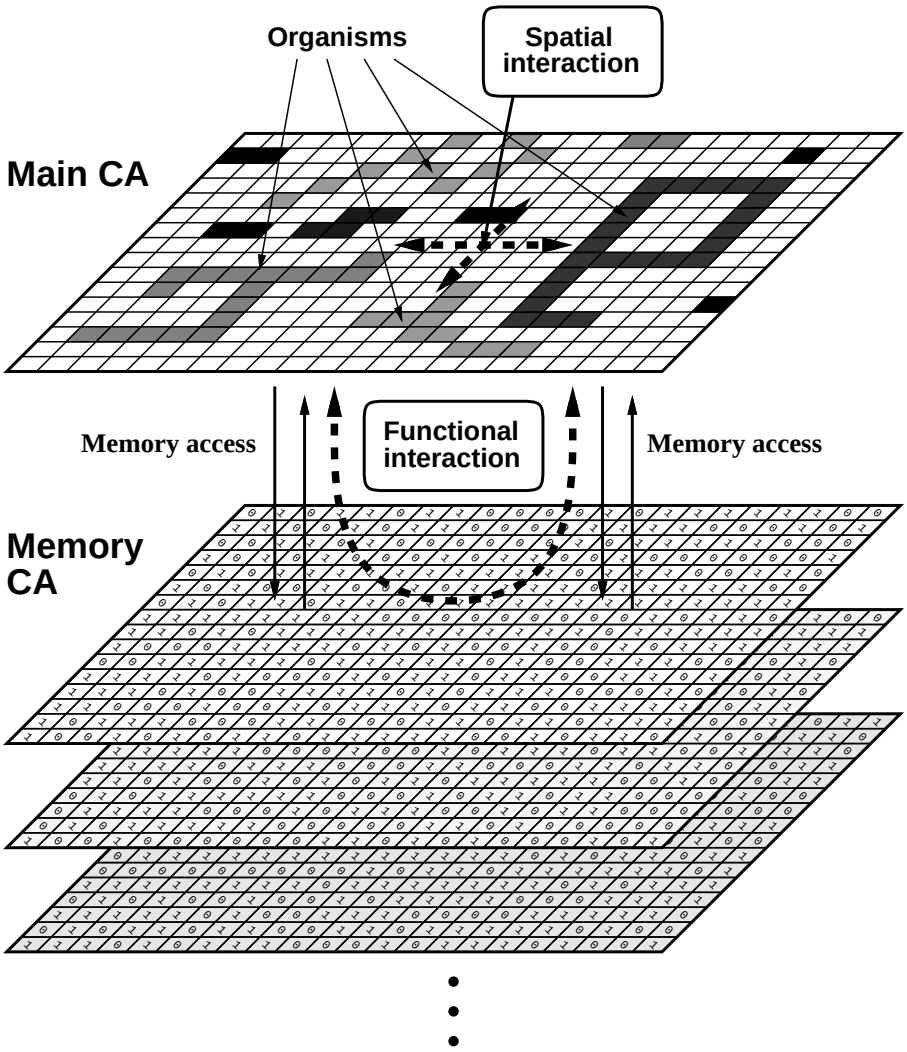


Figure 8.2: Image of organisms functionally interacting with each other through the "memory" CA. The organisms in this CA would be equipped with the ability to read/write information in the memory CA, so they would become able to interact with each other through the access to the same memory CA.

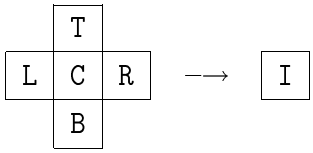
Appendix A

STATE-TRANSITION RULES OF LANGTON'S SR LOOP

The rule tables listed here are written in the following format:

CTRBL->I
00000->0
00001->2
...
...

Each situation `CTRBL` and its image `I` listed in this table are read as follows:



Note that rotations of each situation are not listed in these tables.

All the state-transition rules introduced here are also attached to the simulator softwares which are available from the author's WWW page:

`http://proton.is.s.u-tokyo.ac.jp/~sayama/sdsr/`

Table A.1: State-transition rules of Langton's SR loop.

CTRBL->I	CTRBL->I	CTRBL->I	CTRBL->I	CTRBL->I	CTRBL->I
00000->0	01472->1	11124->4	20032->6	20622->2	40252->0
00001->2	01625->1	11125->1	20042->3	20672->2	40322->1
00002->0	01722->1	11126->1	20051->7	20712->2	50002->2
00003->0	01725->5	11127->7	20052->2	20722->2	50021->5
00005->0	01752->1	11152->2	20057->5	20742->2	50022->5
00006->3	01762->1	11212->1	20072->2	20772->2	50023->2
00007->1	01772->1	11222->1	20102->2	21122->2	50027->2
00011->2	02527->1	11224->4	20112->2	21126->1	50052->0
00012->2	10001->1	11225->1	20122->2	21222->2	50202->2
00013->2	10006->1	11227->7	20142->2	21224->2	50212->2
00021->2	10007->7	11232->1	20172->2	21226->2	50215->2
00022->0	10011->1	11242->4	20202->2	21227->2	50222->0
00023->0	10012->1	11262->1	20203->2	21422->2	50224->4
00026->2	10021->1	11272->7	20205->2	21522->2	50272->2
00027->2	10024->4	11322->1	20207->3	21622->2	51212->2
00032->0	10027->7	12224->4	20212->2	21722->2	51222->0
00052->5	10051->1	12227->7	20215->2	22227->2	51242->2
00062->2	10101->1	12243->4	20221->2	22244->2	51272->2
00072->2	10111->1	12254->7	20222->2	22246->2	60001->1
00102->2	10124->4	12324->4	20227->2	22276->2	60002->1
00112->0	10127->7	12327->7	20232->1	22277->2	60212->0
00202->0	10202->6	12425->5	20242->2	30001->3	61212->5
00203->0	10212->1	12426->7	20245->2	30002->2	61213->1
00205->0	10221->1	12527->5	20252->0	30004->1	61222->5
00212->5	10224->4	20001->2	20255->2	30007->6	70007->7
00222->0	10226->3	20002->2	20262->2	30012->3	70112->0
00232->2	10227->7	20004->2	20272->2	30042->1	70122->0
00522->2	10232->7	20007->1	20312->2	30062->2	70125->0
01232->1	10242->4	20012->2	20321->6	30102->1	70212->0
01242->1	10262->6	20015->2	20322->6	30122->0	70222->1
01252->5	10264->4	20021->2	20342->2	30251->1	70225->1
01262->1	10267->7	20022->2	20422->2	40112->0	70232->1
01272->1	10271->0	20023->2	20512->2	40122->0	70252->5
01275->1	10272->7	20024->2	20521->2	40125->0	70272->0
01422->1	10542->7	20025->0	20522->2	40212->0	
01432->1	11112->1	20026->2	20552->1	40222->1	
01442->1	11122->1	20027->2	20572->5	40232->6	

Appendix B

STATE-TRANSITION RULES OF THE SDSR LOOP

The state-transition rules of the SDSR loop can be obtained by the operations below:

1. Define Langton's rules listed in Table A.1 and their rotationally symmetric ones.
2. To all the undefined situations, apply the following:
   (a) Let 0→1 if it is in the tube and next to 1. Let all other 0s remain as is.
   (b) Let 1→7 if it is in the tube and next to 7. Else, let 1→6 if it is in the tube and next to 6. Else, let 1→4 if it is in the tube and next to 4.
   (c) Let 4,6,7→0 if it is in the tube and next to 0.
   (d) Let 2→1 if it is next to 3. Else, let 2 remain as is if it is next to another 2.

   The criterion for judgment whether a site is in the tube or not is whether at least two sites in the state 1, 2, 4, 6 or 7 are included in its four neighborhood (TRBL).
3. Let 8→0 with no condition.
4. To all the undefined situations in whose four neighborhood (TRBL) there is at least one site in state 8, apply the following:
   (a) Let 0,1→8 if there is at least one site in state 2,3,...,7 in its four neighborhood (TRBL), otherwise let 0→0 and 1→1.
   (b) Let 2,3,5→0.
   (c) Let 4,6,7→1.
5. Define rule '11152->8' and its rotationally symmetric ones.
6. Clear up all the undefined situations by letting 0→0 and 1,2,...,7→8.

Appendix C

STATE-TRANSITION RULES OF THE EVOLOOP

The state-transition rules of the *evoloop* can be obtained by the operations below:

1. Define the rules listed in Table C.1 and their rotationally symmetric ones.
2. Let 8→0 with no condition.
3. To all the undefined situations in whose four neighborhood (TRBL) there is at least one site in state 8, apply the following:
   (a) Let 0,1→8 if there is at least one site in state 2,3,...,7 in its four neighborhood (TRBL), otherwise let 0→0 and 1→1.
   (b) Let 2,3,5→0.
   (c) Let 4,6,7→1.
4. Clear up all the undefined situations by letting 0→0 and 1,2,...,7→8.

Table C.1: Principal part of the state-transition rules of the *evoloop*.

CTRBL->I	CTRBL->I	CTRBL->I	CTRBL->I	CTRBL->I	CTRBL->I
00001->2	10202->1	11272->7	20172->2	21322->2	40125->0
00004->3	10211->1	11273->5	20202->2	21422->2	40162->0
00012->2	10212->1	11322->1	20203->2	21622->2	40212->0
00015->2	10213->1	11332->1	20205->2	21722->2	40215->0
00021->2	10221->1	11542->4	20206->5	22224->2	40222->1
00024->2	10224->4	11572->7	20207->3	22227->2	40232->1
00042->2	10227->7	11624->4	20212->2	22234->2	40262->6
00045->2	10232->4	11627->7	20215->2	22237->2	40312->0
00075->2	10241->4	12224->4	20221->2	22243->2	40322->1
00102->2	10242->4	12227->7	20222->2	22244->2	50002->5
00214->1	10243->4	12243->4	20223->2	22273->2	50012->5
00217->1	10251->1	12273->7	20232->3	22277->2	50021->5
00232->2	10252->7	12324->4	20242->2	22324->3	50023->2
01122->1	10254->3	12327->7	20245->2	22327->3	50024->5
01212->1	10257->7	12426->6	20252->5	30001->3	50027->5
01232->1	10271->7	12433->3	20262->0	30002->2	50042->5
01242->1	10272->7	12627->6	20265->0	30003->2	50072->5
01245->1	10273->5	20001->2	20272->2	30004->3	50202->2
01252->6	10512->1	20002->2	20275->2	30007->4	50205->2
01262->6	10542->4	20004->2	20312->2	30012->3	50212->5
01272->1	10572->7	20005->2	20322->2	30032->2	50215->2
01275->1	10621->1	20006->0	20342->2	30042->1	50242->5
01342->1	10624->4	20007->1	20345->2	30102->1	50272->5
01372->1	10627->7	20012->2	20372->2	30125->0	50312->0
01422->1	11112->1	20015->2	20412->2	30212->3	60202->2
01425->1	11122->1	20021->2	20422->2	30242->3	60212->2
01432->1	11124->4	20022->2	20442->2	30252->1	60222->0
01435->1	11125->1	20023->2	20512->2	30272->3	60242->2
01442->1	11127->7	20024->2	20542->5	30332->1	60272->2
01462->1	11162->1	20026->0	20572->5	31212->3	61222->0
01722->1	11212->1	20027->2	20612->5	31242->3	62224->0
01725->1	11213->1	20032->4	20621->2	31252->1	62227->0
01756->1	11215->1	20042->3	20642->5	31272->3	70102->0
01762->1	11222->1	20045->2	20672->5	32424->3	70112->0
01772->1	11224->4	20054->5	20712->2	32425->1	70122->0
10001->1	11227->7	20057->5	20722->2	32427->3	70125->0
10012->1	11232->1	20062->0	20772->2	32527->1	70162->0
10021->1	11242->4	20072->2	21122->2	32727->3	70212->0
10024->4	11243->4	20075->2	21222->2	40000->1	70215->0
10027->7	11252->7	20102->2	21223->2	40002->1	70222->1
10121->1	11254->3	20112->2	21224->2	40102->0	70232->0
10124->4	11257->7	20122->2	21227->2	40112->0	70262->6
10127->7	11262->6	20142->2	21232->3	40122->0	70312->0

Appendix D

BREAKDOWN OF STATE-TRANSITION RULES OF THE EVOLOOP ACCORDING TO THEIR FUNCTIONS

The principal part of state-transition rules of the *evoloop* listed in the previous chapter is categorized here into six phases and other general rules. Every rule included in six phases is accompanied with a brief comment about its function, which may help readers to imagine the behaviors of signals in CA.

D.1 Phase 1: To lengthen the construction arm

```
20007->1 ; breaking of the tip when hit by gene '7'
70222->1 ; disappearance of gene '7' at the tip

10001->1 ; sustenance of core '1' at the broken tip

00001->2 ; sheathing of the tip
00012->2 ; sheathing of the tip
00021->2 ; sheathing of the tip
00102->2 ; sheathing of the tip touching the root
```

D.2 Phase 2: To turn the tip of the arm left

```
20042->3 ; appearance of left-indicator ‘3’ when hit once by gene ‘4’
40222->1 ; disappearance of the first gene ‘4’ at the tip

11322->1 ; propagation of genes near left-indicator ‘3’
12243->4 ; propagation of genes near left-indicator ‘3’
12273->7 ; propagation of genes near left-indicator ‘3’

20203->2 ; sustenance of sheath ‘2’ next to left-indicator ‘3’
20312->2 ; sustenance of sheath ‘2’ next to left-indicator ‘3’
20342->2 ; sustenance of sheath ‘2’ next to left-indicator ‘3’
20372->2 ; sustenance of sheath ‘2’ next to left-indicator ‘3’
30012->3 ; sustenance of left-indicator ‘3’

30042->1 ; breaking of the side of the tip when hit twice by gene ‘4’
40322->1 ; disappearance of the second gene ‘4’ at the tip

10012->1 ; sustenance of core ‘1’ at the broken tip
```

D.3 Phase 3: To bond the tip and the root of the arm together

```
20207->3 ; appearance of bonder ‘3’ when hit by gene ‘7’

00232->2 ; sheathing of the side of connection next to bonder ‘3’

20232->3 ; bonder ‘3’ moving onto the lower site
21232->3 ; bonder ‘3’ moving onto the lower site
22324->3 ; bonder ‘3’ moving onto the lower site
22327->3 ; bonder ‘3’ moving onto the lower site
30102->1 ; bonder ‘3’ moving onto the lower site

01342->1 ; propagation of genes near bonder ‘3’
01372->1 ; propagation of genes near bonder ‘3’
10213->1 ; propagation of genes near bonder ‘3’
10243->4 ; propagation of genes near bonder ‘3’
11213->1 ; propagation of genes near bonder ‘3’
```

```
11243->4 ; propagation of genes near bonder ‘3’
40312->0 ; propagation of genes near bonder ‘3’
70312->0 ; propagation of genes near bonder ‘3’

20223->2 ; sustenance of sheath ‘2’ next to bonder ‘3’
20322->2 ; sustenance of sheath ‘2’ next to bonder ‘3’
21223->2 ; sustenance of sheath ‘2’ next to bonder ‘3’
21322->2 ; sustenance of sheath ‘2’ next to bonder ‘3’
22234->2 ; sustenance of sheath ‘2’ next to bonder ‘3’
22237->2 ; sustenance of sheath ‘2’ next to bonder ‘3’
22243->2 ; sustenance of sheath ‘2’ next to bonder ‘3’
22273->2 ; sustenance of sheath ‘2’ next to bonder ‘3’

30212->3 ; sustenance of bonder ‘3’
30242->3 ; sustenance of bonder ‘3’
30272->3 ; sustenance of bonder ‘3’
31212->3 ; sustenance of bonder ‘3’
31242->3 ; sustenance of bonder ‘3’
31272->3 ; sustenance of bonder ‘3’
32424->3 ; sustenance of bonder ‘3’
32427->3 ; sustenance of bonder ‘3’
32727->3 ; sustenance of bonder ‘3’

10273->5 ; appearance of messenger ‘5’ onto gene ‘7’
11273->5 ; appearance of messenger ‘5’ onto gene ‘7’

30252->1 ; disappearance of bonder ‘3’ next to messenger ‘5’
31252->1 ; disappearance of bonder ‘3’ next to messenger ‘5’
32425->1 ; disappearance of bonder ‘3’ next to messenger ‘5’
32527->1 ; disappearance of bonder ‘3’ next to messenger ‘5’

01252->6 ; appearance of dissolver ‘6’ out of messenger ‘5’
10252->7 ; recovery of gene ‘7’ from messenger ‘5’
11252->7 ; recovery of gene ‘7’ from messenger ‘5’
20252->5 ; messenger ‘5’ going through onto the sheath
50312->0 ; messenger ‘5’ going through onto the sheath

01756->1 ; propagation of core ‘1’ in the bonding T-junction
```

```
60202->2 ; recovery of sheath ‘2’ behind back-propagating dissolver ‘6’
60212->2 ; recovery of sheath ‘2’ behind back-propagating dissolver ‘6’
60242->2 ; recovery of sheath ‘2’ behind back-propagating dissolver ‘6’
60272->2 ; recovery of sheath ‘2’ behind back-propagating dissolver ‘6’
```

D.4 Phase 4: To dissolve the umbilical cord between parent and offspring

```
20621->2 ; sustenance of sheath ‘2’ on the surface of the offspring
21622->2 ; sustenance of sheath ‘2’ on the surface of the offspring

20262->0 ; dissolution of sheath ‘2’ between offspring and umbilical cord
20265->0 ; dissolution of sheath ‘2’ between offspring and umbilical cord

20006->0 ; dissolution of sheath ‘2’ next to dissolver ‘6’
20026->0 ; dissolution of sheath ‘2’ next to dissolver ‘6’
20062->0 ; dissolution of sheath ‘2’ next to dissolver ‘6’

01262->6 ; back-propagation of dissolver ‘6’
11262->6 ; back-propagation of dissolver ‘6’
12426->6 ; back-propagation of dissolver ‘6’
12627->6 ; back-propagation of dissolver ‘6’
40262->6 ; back-propagation of dissolver ‘6’
70262->6 ; back-propagation of dissolver ‘6’

60222->0 ; disappearance of dissolver ‘6’ at the end of the umbilical cord
61222->0 ; disappearance of dissolver ‘6’ at the end of the umbilical cord
62224->0 ; disappearance of dissolver ‘6’ at the end of the umbilical cord
62227->0 ; disappearance of dissolver ‘6’ at the end of the umbilical cord

01462->1 ; propagation of genes at the T-junction with dissolver ‘6’
01762->1 ; propagation of genes at the T-junction with dissolver ‘6’
10621->1 ; propagation of genes at the T-junction with dissolver ‘6’
10624->4 ; propagation of genes at the T-junction with dissolver ‘6’
10627->7 ; propagation of genes at the T-junction with dissolver ‘6’
11162->1 ; propagation of genes at the T-junction with dissolver ‘6’
11624->4 ; propagation of genes at the T-junction with dissolver ‘6’
```

```
11627->7 ; propagation of genes at the T-junction with dissolver ‘6’
40162->0 ; propagation of genes at the T-junction with dissolver ‘6’
70162->0 ; propagation of genes at the T-junction with dissolver ‘6’

20206->5 ; appearance of messenger ‘5’ out of dissolver ‘6’
20612->5 ; appearance of messenger ‘5’ out of dissolver ‘6’
20642->5 ; appearance of messenger ‘5’ out of dissolver ‘6’
20672->5 ; appearance of messenger ‘5’ out of dissolver ‘6’

50012->5 ; messenger ‘5’ waiting for genes coming at the broken corner
50042->5 ; messenger ‘5’ waiting for genes coming at the broken corner
50072->5 ; messenger ‘5’ waiting for genes coming at the broken corner

00214->1 ; propagation of genes at the broken corner
00217->1 ; propagation of genes at the broken corner
10021->1 ; propagation of genes at the broken corner
10024->4 ; propagation of genes at the broken corner
10027->7 ; propagation of genes at the broken corner
10211->1 ; propagation of genes at the broken corner
10241->4 ; propagation of genes at the broken corner
10271->7 ; propagation of genes at the broken corner
40102->0 ; propagation of genes at the broken corner
70102->0 ; propagation of genes at the broken corner

00015->2 ; recovery of sheath ‘2’ at the broken corner
00045->2 ; recovery of sheath ‘2’ at the broken corner
00075->2 ; recovery of sheath ‘2’ at the broken corner

50212->5 ; messenger ‘5’ waiting for genes coming at the corner
50242->5 ; messenger ‘5’ waiting for genes coming at the corner
50272->5 ; messenger ‘5’ waiting for genes coming at the corner

20005->2 ; sustenance of sheath ‘2’ next to starting messenger ‘5’
20015->2 ; sustenance of sheath ‘2’ next to starting messenger ‘5’
20045->2 ; sustenance of sheath ‘2’ next to starting messenger ‘5’
20075->2 ; sustenance of sheath ‘2’ next to starting messenger ‘5’
```

D.5 Phase 5: To germinate a new sprout of the arm

```
01245->1 ; propagation of genes in sheath structures with messenger ‘5’
01275->1 ; propagation of genes in sheath structures with messenger ‘5’
10512->1 ; propagation of genes in sheath structures with messenger ‘5’
10542->4 ; propagation of genes in sheath structures with messenger ‘5’
10572->7 ; propagation of genes in sheath structures with messenger ‘5’
11215->1 ; propagation of genes in sheath structures with messenger ‘5’
11542->4 ; propagation of genes in sheath structures with messenger ‘5’
11572->7 ; propagation of genes in sheath structures with messenger ‘5’
40215->0 ; propagation of genes in sheath structures with messenger ‘5’
70215->0 ; propagation of genes in sheath structures with messenger ‘5’

20205->2 ; sustenance of sheath ‘2’ next to messenger ‘5’
20215->2 ; sustenance of sheath ‘2’ next to messenger ‘5’
20245->2 ; sustenance of sheath ‘2’ next to messenger ‘5’
20275->2 ; sustenance of sheath ‘2’ next to messenger ‘5’
20345->2 ; sustenance of sheath ‘2’ next to messenger ‘5’
20512->2 ; sustenance of sheath ‘2’ next to messenger ‘5’

20542->5 ; propagation of messenger ‘5’ on sheath structures
20572->5 ; propagation of messenger ‘5’ on sheath structures

50202->2 ; recovery of sheath ‘2’ from messenger ‘5’
50205->2 ; recovery of sheath ‘2’ from messenger ‘5’
50215->2 ; recovery of sheath ‘2’ from messenger ‘5’

20054->5 ; messenger ‘5’ arriving at the next corner
20057->5 ; messenger ‘5’ arriving at the next corner

50002->5 ; messenger ‘5’ waiting for gene ‘4’ coming at the corner
50021->5 ; messenger ‘5’ waiting for gene ‘4’ coming at the corner
50024->5 ; messenger ‘5’ waiting for gene ‘4’ coming at the corner
50027->5 ; messenger ‘5’ waiting for gene ‘4’ coming at the corner

01425->1 ; propagation of genes at the corner with messenger ‘5’
01435->1 ; propagation of genes at the corner with messenger ‘5’
01725->1 ; propagation of genes at the corner with messenger ‘5’
```

```
10251->1 ; propagation of genes at the corner with messenger ‘5’
10257->7 ; propagation of genes at the corner with messenger ‘5’
11125->1 ; propagation of genes at the corner with messenger ‘5’
11257->7 ; propagation of genes at the corner with messenger ‘5’
40125->0 ; propagation of genes at the corner with messenger ‘5’
70125->0 ; propagation of genes at the corner with messenger ‘5’

10254->3 ; appearance of sprout-generator ‘3’ at the corner
11254->3 ; appearance of sprout-generator ‘3’ at the corner

01232->1 ; propagation of core ‘1’ behind sprout-generator ‘3’
10232->4 ; recovery of gene ‘4’ from sprout-generator ‘3’
20023->2 ; sustenance of sheath ‘2’ next to sprout-generator ‘3’
20032->4 ; appearance of sprout-guide ‘4’ from sprout-generator ‘3’
30125->0 ; disappearance of sprout-generator ‘3’
50023->2 ; disappearance of messenger ‘5’ next to sprout-generator ‘3’

40002->1 ; disappearance of sprout-guide ‘4’
```

D.6 Phase 6: To lengthen the new sprout of the arm

```
00004->3 ; appearance of sprout-capper ‘3’ from sprout-guide ‘4’
00024->2 ; sheathing of the tip of the sprout
00042->2 ; sheathing of the tip of the sprout
40000->1 ; disappearance of sprout-guide ‘4’

11232->1 ; propagation of genes at the tip of the sprout
11332->1 ; propagation of genes at the tip of the sprout
12324->4 ; propagation of genes at the tip of the sprout
12327->7 ; propagation of genes at the tip of the sprout
40232->1 ; disappearance of gene ‘4’ at the tip of the sprout
70232->0 ; disappearance of gene ‘7’ at the tip of the sprout

30001->3 ; sustenance of sprout-capper ‘3’
30004->3 ; sustenance of sprout-capper ‘3’
30007->4 ; breaking of the tip of the sprout when hit by gene ‘7’

12433->3 ; alteration of the second gene ‘4’ into sprout-finisher ‘3’
```

```
30003->2 ; alteration of sprout-capper ‘3’ into sheath ‘2’
30032->2 ; alteration of left-indicator ‘3’ into sheath ‘2’
30332->1 ; disappearance of sprout-finisher ‘3’
```

D.7 General rules for the sustenance of sheath structures and the propagation of genes

In straight sheath structures

```
01212->1
01242->1
01272->1
10202->1
10212->1
10242->4
10272->7
11212->1
11242->4
11272->7
40212->0
70212->0

20202->2
20212->2
20242->2
20272->2
```

At the corner of sheath structures

```
01122->1
01422->1
01432->1 ; propagation of core ‘1’ at the corner with left-indicator ‘3’
01722->1
10221->1
10224->4
10227->7
11122->1
11224->4
```

```
11227->7
40122->0
70122->0

20002->2
20012->2
20021->2
20022->2
20024->2
20027->2
20072->2
20122->2
20221->2
20422->2
20722->2
21122->2
21224->2
21227->2
21422->2
21722->2
30002->2 ; alteration of left-indicator ‘3’ at the corner into sheath ‘2’
```

At the T-junction of sheath structures

```
01442->1
01772->1
10121->1
10124->4
10127->7
11112->1
11124->4
11127->7
40112->0
70112->0

22244->2
22277->2
20112->2 ; at the T-junction near the sprout which has just germinated
20442->2 ; at the T-junction near the sprout which has just germinated
```

```
20772->2 ; at the T-junction near the sprout which has just germinated
```

At the tip of the construction arm

```
11222->1
12224->4
12227->7

20001->2
20004->2
20102->2 ; at the tip which has just bended
20142->2 ; at the tip which has just bended
20172->2 ; at the tip which has just bended
20412->2 ; at the tip which has just bended
20712->2 ; at the tip which has just bended
20222->2 ; at the tip touching the root
21222->2 ; at the tip touching the root
22224->2 ; at the tip touching the root
22227->2 ; at the tip touching the root
```

Bibliography

[1] Adami, C. and Brown, C. T. : Evolutionary learning in the 2D artificial life system "Avida", in Brooks, R. and Maes, P. eds., *Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*, pp. 377–381, Cambridge, Massachusetts (1994), MIT Press.

[2] Adleman, L. M. : Molecular computation of solutions to combinatorial problems, *Science*, **266** pp. 1021–1024 (1994).

[3] Arbib, M. A. : A simple self-reproducing universal automaton, *Information and Control*, **9** pp. 177–189 (1966).

[4] Bagnoli, F., Rechtman, R., and Ruffo, S. : Some facts of life, *Physica A*, **171** pp. 249–264 (1991).

[5] Berlekamp, E. R., Conway, J. H., and Guy, R. K. : *Winning Ways*, volume 2, pp. 817–850, Academic Press, New York (1982).

[6] Burks, A. W. ed. : *Essays on Cellular Automata*, University of Illinois Press, Urbana, Illinois (1970).

[7] Burks, A. W. : Von Neumann's self-reproducing automata, in Burks, A. W. ed., *Essays on Cellular Automata*, pp. 3–64, University of Illinois Press, Urbana, Illinois (1970).

[8] Byl, J. : Self-reproduction in small cellular automata, *Physica D*, **34** pp. 295–299 (1989).

[9] Chou, H. H. and Reggia, J. A. : Emergence of self-replicating structures in a cellular automata space, *Physica D*, **110** pp. 252–276 (1997).

[10] Chou, H. H. and Reggia, J. A. : Problem solving during artificial selection of self-replicating Loops, *Physica D*, **115** pp. 293–312 (1998).

[11] Codd, E. F. : *Cellular Automata*, ACM Monograph Series, Academic Press, New York (1968).

[12] Dewdney, A. K. : In the game called Core War hostile programs engage in a battle of bits, *Scientific American*, **250** No. 5, pp. 15–19 (1984).

[13] Dewdney, A. K. : A Core War bestiary of viruses, worms and other threats to computer memories, *Scientific American*, **252** No. 3, pp. 14–19 (1985).

[14] Frisch, U., Hasslacher, B., and Pomeau, Y. : Lattice-gas automata for the Navier-Stokes Equation, *Physical Review Letters*, **56** pp. 1505–1508 (1986).

[15] Garcia, J. B. C., Gomes, M. A. F., Jyh, T. I., Ren, T. I., and Sales, T. R. M. : Nonlinear dynamics of the cellular-automaton "game of life", *Physical Review E*, **48** pp. 3345–3351 (1993).

[16] Gardner, M. : The fantastic combinations of John Conway's new solitaire game Life, *Scientific American*, **223** No. 4, pp. 120–123 (1970).

[17] Gardner, M. : On cellular automata, self-reproduction, the Garden of Eden and the game Life, *Scientific American*, **224** No. 2, pp. 112–117 (1971).

[18] Gers, F. and de Garis, H. : CAM-Brain: A new model for ATR's cellular automata based artificial brain project, in Higuchi, T., Iwata, M., and Liu, W. eds., *Evolvable Systems: From Biology to Hardware, Proceedings of the First International Conference on Evolvable Systems (ICES'96)*, volume 1259 of *Lecture Notes in Computer Science*, pp. 437–452, Tsukuba, Japan (1996), Springer-Verlag.

[19] Goldberg, D. E. : *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, Massachusetts (1989).

[20] Goodsell, D. S. : *The Machinery of Life*, Springer-Verlag, New York (1993).

[21] Gotts, N. M. and Callahan, P. B. : Emergent structures in sparse fields of Conway's "Game of Life", in Adami, C., Belew, R. K., Kitano, H., and Taylor, C. E. eds., *Artificial Life VI: Proceedings of the Sixth International Conference on Artificial Life*, pp. 104–113, Los Angeles, California (1998), MIT Press.

[22] Hardy, J., de Pazzis, O., and Pomeau, Y. : Molecular dynamics of a classical lattice gas: Transport properties and time correlation functions, *Physical Review A*, **13** pp. 1949–1960 (1976).

[23] Holland, J. H. : *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, University of Michigan Press, Ann Arbor, Michigan (1975).

[24] Hopcroft, J. E. and Ullman, J. D. : *Formal Languages and Their Relation to Automata*, Addison-Wesley, Reading, Massachusetts (1969).

[25] Kato, Y., Mitsunari, T., and Tsukiyama, H. : *Cellular Automaton Method*, Morikita Syuppan, Tokyo (1998), written in Japanese.

[26] Koza, J. R. : *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, Massachusetts (1992).

[27] Langton, C. G. : Self-reproduction in cellular automata, *Physica D*, **10** pp. 135–144 (1984).

[28] Langton, C. G. : Studying artificial life with cellular automata, *Physica D*, **22** pp. 120–149 (1986).

[29] Langton, C. G. : Artificial life, in *Artificial Life: Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems*, volume VI of *SFI Studies in the Sciences of Complexity*, pp. 1–47, Los Alamos, New Mexico (1987), Addison-Wesley.

[30] Lohn, J. D. and Reggia, J. A. : Automatic discovery of self-replicating structures in cellular automata, *IEEE Transactions on Evolutionary Computation*, **1** No. 3, pp. 165–178 (1997).

[31] Mitchell, M., Crutchfield, J. P., and Hraber, P. T. : Evolving cellular automata to perform computations: Mechanisms and impediments, *Physica D*, **75** pp. 361–391 (1994).

[32] Morita, K. and Imai, K. : Self-reproduction in a reversible cellular space, in *Proceedings of the International Workshop on Universal Machines and Computations*, Paris, France (1995).

[33] Morita, K. and Imai, K. : A simple self-reproducing cellular automaton with shape-encoding mechanism, in Langton, C. G. and Shimohara, K. eds., *Artificial Life V: Proceedings of the Fifth International Workshop on the Synthesis and Simulation of Living Systems*, pp. 489–496, Nara, Japan (1996), MIT Press.

[34] Murray, J. D. : A pre-pattern formation mechanism for animal coat markings, *Journal of Theoretical Biology*, **88** pp. 161–199 (1981).

[35] Oohashi, T., Nakata, D., Kikuta, T., and Murakami, K. : Programmed self-decomposition model, *Journal of the Japan Association for Philosophy of Science*, **18** No. 2, pp. 79–87 (1987), written in Japanese.

[36] Oohashi, T., Sayama, H., Ueno, O., and Maekawa, T. : Artificial life based on programmed self-decomposition model, Technical Report TR-H-198, ATR (1996).

[37] Pargellis, A. N. : The evolution of self-replicating computer organisms, *Physica D*, **98** pp. 111–127 (1996).

[38] Perrier, J. Y., Sipper, M., and Zahnd, J. : Toward a viable, self-reproducing universal computer, *Physica D*, **97** pp. 335–352 (1996).

[39] Pesavento, U. : An implementation of von Neumann's self-reproducing machine, *Artificial Life Journal*, **2** No. 4, pp. 337–354 (1995).

[40] Poundstone, W. : *The Recursive Universe*, William Morrow and Company, New York (1985).

[41] Ray, T. S. : An approach to the synthesis of life, in Langton, C. G., Taylor, C., Farmer, J. D., and Rasmussen, S. eds., *Artificial Life II: Proceedings of the Workshop on Artificial Life*, volume X of *SFI Studies in the Sciences of Complexity*, pp. 371–408, Santa Fe, New Mexico (1990), Addison-Wesley.

[42] Reggia, J. A., Armentrout, S. L., Chou, H. H., and Peng, Y. : Simple systems that exhibit self-directed replication, *Science*, **259** pp. 1282–1287 (1993).

[43] Sayama, H. : Development of highly-parallel simulator of individuals of virtual automata for verification of "programmed self-decomposition model", *IPSJ Transactions*, **39** No. 6, pp. 1782–1789 (1998), written in Japanese.

[44] Sayama, H. : Introduction of structural dissolution into Langton's self-reproducing loop, in Adami, C., Belew, R. K., Kitano, H., and Taylor, C. E. eds., *Artificial Life VI: Proceedings of the Sixth International Conference on Artificial Life*, pp. 114–122, Los Angeles, California (1998), MIT Press.

[45] Sayama, H. : Spontaneous evolution of self-reproducing loops implemented on cellular automata: A preliminary report, in Bar-Yam, Y. ed., *Proceedings of the Second International Conference on Complex Systems*, Nashua, New Hampshire (1998), Perseus Books, forthcoming; also submitted to *InterJournal of Complex Systems*, manuscript number 236.

[46] Sayama, H. : A new structurally dissolvable self-reproducing loop evolving in a simple cellular automata space, submitted (1999).

[47] Sayama, H. : A structurally dissolvable self-reproducing loop implemented from Langton's self-reproducing loop, *IPSJ Transactions*, **40** No. SIG2 (MPS1), (1999), written in Japanese, forthcoming.

[48] Sayama, H. : Toward the realization of an evolving ecosystem on cellular automata, in Sugisaka, M. and Tanaka, H. eds., *Proceedings of the Fourth International Symposium*

on Artificial Life and Robotics (AROB 4th '99), pp. 254–257, Beppu, Oita, Japan (1999).

[49] Sipper, M. : Non-uniform cellular automata: Evolution in rule space and formation of complex structures, in Brooks, R. and Maes, P. eds., *Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*, pp. 394–399, Cambridge, Massachusetts (1994), MIT Press.

[50] Sipper, M. : *Evolution of Parallel Cellular Machines*, volume 1194 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin (1997).

[51] Sipper, M., Mange, D., and Stauffer, A. : Ontogenetic hardware, *BioSystems*, **44** pp. 193–207 (1997).

[52] Sipper, M., Sanchez, E., Mange, D., Tomassini, M., Prez-Uribe, A., and Stauffer, A. : A phylogenetic, ontogenetic, and epigenetic view of bio-inspired hardware systems, *IEEE Transactions on Evolutionary Computation*, **1** No. 1, pp. 83–97 (1997).

[53] Tempesti, G. : A new self-reproducing cellular automaton capable of construction and computation, in Morán, F., Moreno, A., Merelo, J. J., and Chacón, P. eds., *Advances in Artificial Life: Proceedings of the Third European Conference on Artificial Life (ECAL'95)*, volume 929, Lecture Notes in Artificial Intelligence of *Lecture Notes in Computer Science*, pp. 555–563, Granada, Spain (1995), Springer-Verlag.

[54] Todd, P. M. : Artificial death, in Deneubourg, J. L., Bersini, H., Goss, S., Nicolis, G., and Dagonnier, R. eds., *Proceedings of the Second European Conference on Artificial Life (ECAL'93)*, volume 2, pp. 1048–1059, Brussels, Belgium (1993).

[55] Toffoli, T. and Margolus, N. : *Cellular Automata Machines*, MIT Press, Cambridge, Massachusetts (1987).

[56] Turing, A. M. : The chemical basis of morphogenesis, *Philosophical Transactions of the Royal Society of London*, **B 237** pp. 37–72 (1952).

[57] Vitányi, P. M. B. : Sexually reproducing cellular automata, *Mathematical Biosciences*, **18** pp. 23–54 (1973).

[58] von Neumann, J. : *Theory of Self-Reproducing Automata*, University of Illinois Press, Urbana, Illinois (1966), edited and completed by A. W. Burks.

[59] von Neumann, J. : The general and logical theory of automata, in Aspray, W. and Burks, A. W. eds., *Papers of John von Neumann on Computing and Computer Theory*, pp. 391–431, MIT Press, Cambridge, Massachusetts (1987), originally presented in 1948.

[60] Wang, H. : A variant to Turing's theory of computing machines, *Journal of ACM*, **4** pp. 63–92 (1957).

[61] Wolfram, S. : Universality and complexity in cellular automata, *Physica D*, **10** pp. 1–35 (1984).

[62] Young, D. A. : A local activator-inhibitor model of vertebrate skin patterns, *Mathematical Biosciences*, **72** pp. 51–58 (1984).