Truth Table Language for Generating Self-replicating Systems

Hajime Harada and Yukihiko Toquenaga
Integrative Environmental Sciences
University of Tsukuba
Ibaraki 305-8572, Japan
E-mail: hajime@pe.ies.life.tsukuba.ac.jp

Abstract

In Artificial Life studies, a self-replicating system is considered to have mostly perfect copying capacity. However this feature is sensitive external to perturbation. We proposed a simple self-replication system with self-irreplicable individuals. The system consists of truth table relations for generating individual notations. The system incorporating 1-D spatial configuration showed cyclic behavior. Within the cycle, the system acquired replication redundancy, or offspring was of the same species irrespective of which neighboring individual was chosen as a mate. Another feature of this system was its robustness; when one or more individuals were altered by noise effect, the system absorbed the perturbation. Creating a Truth Table Language (TTL) and a self-replicating system is for the purpose of orienting to a platform for bioinformatic research. Dealing with TTL systems as study subjects, we may further propose new techniques for analyzing TTL systems, which can be used as feedback into bioinformatics.

keywords: Artificial Life, truth table, redundancy, cyclic behavior, bioinformatics.

1 Introduction

Most self-replicating systems coined in Artificial Life studies start from one or several ancestral species or individuals that are capable of perfectly copying themselves. In the field of Artificial Life, many experiments, such as artificial chemistry[1, 2], Tierra[3], Avida[4], random access language[5], machine and tape dynamics[6], and binary string system[7, 8], have been performed in order to study self-reproduction systems.

However, these studies focused on perfect replication in self-reproduction. In such systems, perturbation by mutation and other evolutionary operations, which work against the self-copying process, generate lucky variants in species which later prevail within the system. Thus, making perfectly self-replicating ancestral species becomes the key to constructing artificial self-replicating systems. Additionally redundant and robust language should be used for self-replication in order to protect against genetic perturbation so as to prevent the core of the self-replication processes from generating variant species which are not capable of self-replication.

What happens when not all ancestral species are self-replicating, but some of them reproduce in hypercycles? Hypercycle[9] is recognized as the common theory of replication cycles, in which each member catalyzes the replication of the next in the cycle, and may be dynamic in the sense that members of each hypercycle change both temporally and spatially. The coexistence of multiple dynamic hypercycles as an integrated whole can realize a self-replication system. Such a system may be robust and hence resistant to external perturbation, similar to immune systems realized in biological systems. However, several studies have argued that hypercycles seem to be adversely affected by noise or perturbations, causing errors in replication, and that they are unstable in regards to persistence at higher phases[10, 11].

In this article, we propose the Truth Table Language (TTL) for generating a self-replication system that starts with many self-irreplicapable species and a pair of self-replicating species. It has a simple structure, but shows some interesting behavior against external noise.

2 Model

For self- or other-replication, we used bit-matching rules applying 16 possible rules in the truth table (Fig. 1). The system consists of individuals, each of which is coded by a specified length of a binary bit-string (Fig. 2A). For example, in the case of a 6-bit length, there were 64 types of individuals in the system. Pairs were then formed among the individuals for producing offspring. Each individual was able to select one individual to pair with, so that the direction went from one individual to another. Individuals in a pair were defined as "Self" and "Mate" to clarify the direction of producing offspring; the Self individual produced offspring with the Mate individual.

(A)

Self	Mate	Combination of bit-pattern
(1 ,	1)	case 1
(1 ,	0)	case 2
(0 ,	1)	case 3
(0 ,	0)	case 4

(B)

Decimal number	Logic function name	case1 (1, 1)	case2 (1, 0)	case3 (0, 1)	case4 (0, 0)
		Inputs (Self, Mate)			
0	ALL0	0	0	0	0
1	NOR	0	0	0	1
2		0	0	1	0
3		0	0	1	1
4		0	1	0	0
5		0	1	0	1
6	XOR	0	1	1	0
7	NAND	0	1	1	1
8	AND	1	0	0	0
9	XNOR	1	0	0	1
10		1	0	1	0
11		1	0	1	1
12		1	1	0	0
13		1	1	0	1
14	OR	1	1	1	0
15	ALL1	1	1	1	1

Figure 1: Truth Table. (A):2-bit input.(B):16 possible combinations

The bit pattern of an offspring was determined by locus-wise bit-matching between the two aligned bit-strings of the parents (Fig. 2B), and each individual had a head and tail position in the bit pattern. The length of individuals in the system was the same, so that the alignment of two individuals' bit patterns was simply a matter of aligning each individual's head or tail position. After alignment, each locus had a position in one of four possible patterns (00, 01, 10, or 11) (Fig. 1A).

At this point, we considered the bit pattern as input for producing a 1-bit pattern of the offspring's locus. When an offspring's 1-bit pattern is determined using a pair's 2-bit pattern on one locus, there are 16 patterns possible (there are 2 single-bit patterns, and there are four 2-bit patterns for one locus. Thus the combination is $2^4 = 16$). These input and output patterns are just as a 2-state and 2-input truth table (Fig. 1B).

The bit-pattern at a locus (00, 01, 10, or 11) was then decoded with one of the 16 possible truth table functions for 2-bit patterns. To make the truth table, we needed to extract four bits, which were used for deciding the new 1-bit pattern (Fig. 2B). The pattern consisted of four bits, surrounding the target locus, used to determine which truth table was applied for

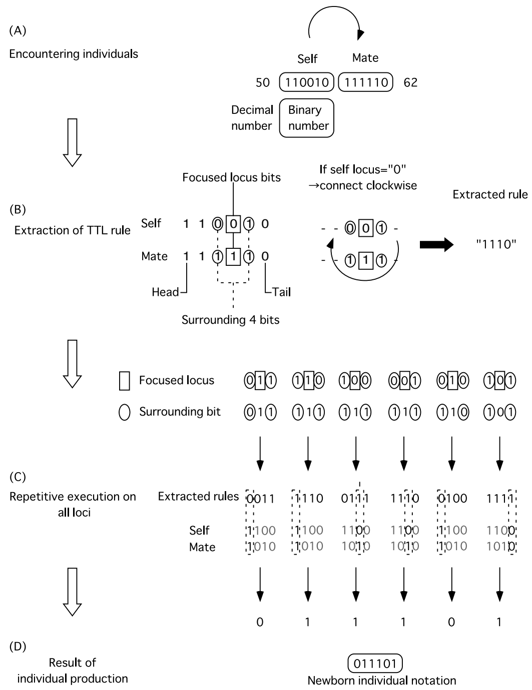

Figure 2: Individual production schema

locus-wise translation. At the locus point, Self's locus bit was the focus for deciding the linking order for extracting four bits. When the focal locus bit was "0", the four bits neighboring the locus were linked in a clockwise sequence, which started from the Self's lower bit, through the Mate's lower and upper bits, and finished at the Self's upper bit. When the Mate's bit was "1", four bits were linked in a counterclockwise sequence. This method, which linked four bits to one sequence, is not biased, and can extract all sorts of 16 rules on the truth table. An extra step was applied only to the loci in the head and tail positions. In these positions, the bit sequence of an individual was defined as a circular structure, so that the bits of head and tail loci were defined as adjacent bits for the execution of the above extraction methods. When the head locus was focal, the tail locus was considered adjacent, and vice versa. Thus, the production of a new individual was completely deterministic and depended solely on the bit-patterns of the two parent individuals.

3 TTL Systems

In this system, exactly one offspring was generated at each reproduction event. The parent individual was then replaced by the newborn offspring, and hence, the total number of individuals was kept constant. As each individual had a fixed length, the number of pairings between Self and Mate individuals was finite. For example, the number of pairings (sorts) between 6-bit individuals was $4,096(2^6 \times 2^6)$, because of the existence of a one-way production direction for Self individuals. As can be seen by the absence of a grayscale line running diagonally from the top left to the bottom right of the diagrams in Figure 3, perfectly replicating individuals did not, for the most part, exist within the system. Only the pairs of each 00...0s and 11...1s individuals were able to produce offspring identical to the Self and Mate individuals' bit strings. A few pairs were able to reproduce an individual identical to Self, though these were not perfect copies in the sense that Mate's bit string was not identical to the offspring's, and vice versa.

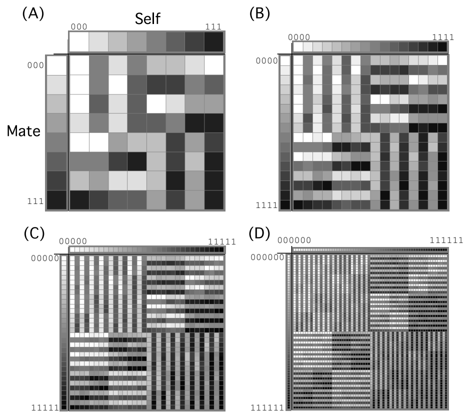

Figure 3: The matrixes of individual combinations: Individuals ordered as binary digits are suggested as grayscale. (00...0): white, (11...1): black. Each matrix shows the individual length: (A)3, (B)4, (C)5, and(D)6.

If there was no spatial structure (Fig. 4A), a small number of bit-patterns (species) were able to persist. After a transient phase, the system dynamics converged into cyclic behavior; each location in the loop was occupied by a specific series of individual species. The word "convergence" does not mean a stable state in which an individual is producing an identical individual with same bit-string, but rather that the group, which consists of specific individuals, is sequentially producing individuals of each other, in which case the dynamics of production by the group are closed. Thus, we incorporated a 1-D spatial constraint; individuals were arranged in a loop and a reproducing individual interacted with one of its one-step neighbors (Fig. 4B).

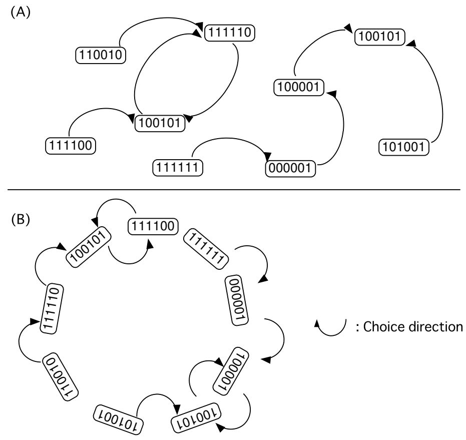

Figure 4: Spatial settings. (A): non-spatial. (B): 1-D spatial restriction

4 Results

The results obtained by incorporating 1-D spatial restrictions are as follows. The composition of individuals in the loop at the state of convergence was not estimated from these initial compositions. Nevertheless the production process by the parents is predictable. Although the initial configuration strongly influenced the system, the diversity of emerging individuals was maintained.

The cyclic behavior of individual production was revealed either when all individuals in the loop had the same bit-string, or when a Self individual was able to reproduce an individual with a bit-string matching that of the Mate, irregardless of whether or not it matched the Self's bit-string. Within the cycle, the same species was reproduced as an offspring irrespective of which neighboring individual was chosen as Mate (Fig. 5).

Another feature of the system with 1-D restrictions is its robustness. When one or more individuals were replaced by randomly selected individuals of a different species, the system absorbed the perturbation. Individual replacement is the same as noise injection directly onto a bit-string (Fig. 6). In Figure 6, after the first replacement occurred, the inserted individual reproduced a bit-string different from that of the previous series. Replacement did not seem to have a wide range of effects in the loop domain. The loop structure has absorbing capabilities. This behavior influences the phenomenon of early convergent generations. Thus, the loop of individuals achieved a stable cyclic transition with redundant reproduction, and proved to be robust in regards to external perturbation.

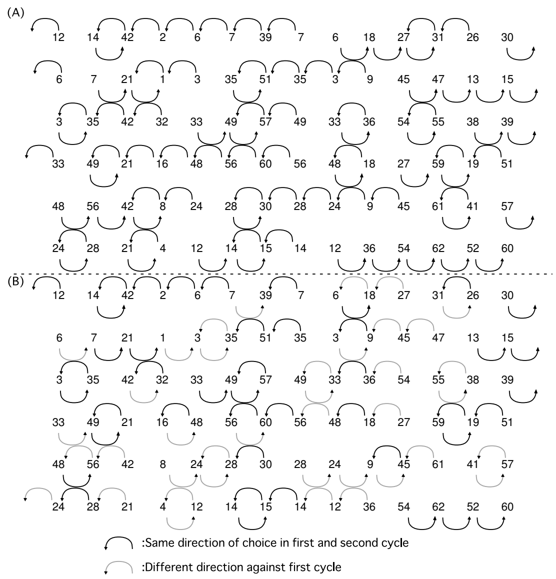

Figure 5: The result of selection behavior. (A): First cycle. (B): Second cycle. Despite random individual selection, each cycle has the same individual production.

5 Discussion

The proposed TTL system with 1-D spatial structure showed robustness in regards to external perturbation and redundant reproduction as a whole. However, it was not our intent that these phenomena emerge during the process of individual production. The structure of each individual as well as the TTL system applied to individual production are simple, and are not intended to intrinsically replicate perfectly.

We may be able to extract one generation from the results of the 1-D restriction system, and consider it as a coded biological string, such as a DNA string, by transposing the string into a DNA sequence and analyzing it with bioinformatic tools. If meaningful results are produced, we must ask ourselves how they should be interpreted. These bioinformatic tools merely suggest the homology between the strings and the character data coded by the base (A, T, G, and C). They

Individuals on 1-D restricted space

Generations	Loop structure									
149	13	9	0	36	45	37	52	38	54	51
150	38	36	0	18	54	50	26	19	27	57
151	19	18	0	9	27	25	13	41	45	60
152	41	9	0	36	45	44	38	52	54	30
153	52	36	0	18	54	22	19	26	27	15
154	26	18	0	9	27	11	41	13	45	39
155	13	9	0	36	45	37	52	38	54	51
156	38	36	0	18	54	50	26	19	27	57
157	19	18	0	9	27	25	13	41	45	60
158	41	9	0	36	45	44	38	52	54	30
159	52	36	0	18	54	22	19	26	27	15
160	26	18	0	9	27	11	41	13	45	39
161	13	9	<31>	36	45	37	52	38	54	51
162	38	36	(45)	(43)	54	50	26	19	27	57
163	19	18	[54]	(57)	(23)	25	13	41	45	60
164	41	9	[27]	(58)	(45)	44	38	52	54	30
165	52	36	[45]	(61)	(53)	22	19	26	27	15
166	26	18	[54]	[62]	[58]	11	41	13	45	39
167	13	9	[27]	[31]	(45)	(21)	52	38	54	51
168	38	36	[45]	[47]	(46)	(10)	(58)	19	27	57
169	19	18	[54]	[55]	[23]	(21)	(61)	(9)	45	60
170	41	9	27	[59]	[43]	[42]	[62]	52	54	30
171	52	36	45	[61]	[53]	[21]	[31]	26	27	15
172	26	18	54	[62]	[58]	[42]	[47]	13	45	39
173	13	9	27	31	[29]	[21]	[55]	38	54	51
174	38	36	45	47	[46]	[42]	[59]	19	27	57
175	19	18	54	55	23	[21]	[61]	41	45	60
176	41	9	27	59	43	42	62	52	54	30
177	52	36	45	61	53	21	31	26	27	15
178	26	18	54	62	58	42	47	13	45	39
179	13	9	27	31	29	21	55	38	54	51
180	38	36	45	47	46	42	59	19	27	57
181	19	18	54	55	23	21	61	41	45	60
182	41	9	27	59	43	42	62	52	54	30

< >: Modified individual by external noise(or replaced indiviual)
( ): Affected newborn individuals by the neighborhood replacement
[ ]: Newborn individuals cyclic behavior produced by noise-affected individuals

Figure 6: Noise effect on 1-D spatial restriction.

show neither the informational relation between the function of the TTL system nor the biological meanings between an artificial string and a biological string. If these bioinformatic tools suggest the homology of some genes, they may find some "biased" patterns affected through production within a TTL system (Fig. 7).

What is the sequence pattern? How does the pattern emerge? The bioinformatic tools are considered to "··· look for patterns and make predictions without a complete understanding of where biological data comes from and what it means[12](p.4)." However the supreme goal of genome sciences should be to solve the algorithms that realize homologous or similar sequence patterns in the comparisons between extracted DNA data.

6 Future Direction

The TTL system is limited in fixed individual length for individual production, and so requires system improvements. Creating the TTL and a self-replicating system is not our final goal. In this paper, we proposed the present self-replicating system as a platform for bioinformatic research.

- Generally, one can only infer the evolutionary events from real biological data. However, the present system includes true trees and lineages for specific evolutionary events, and all histories

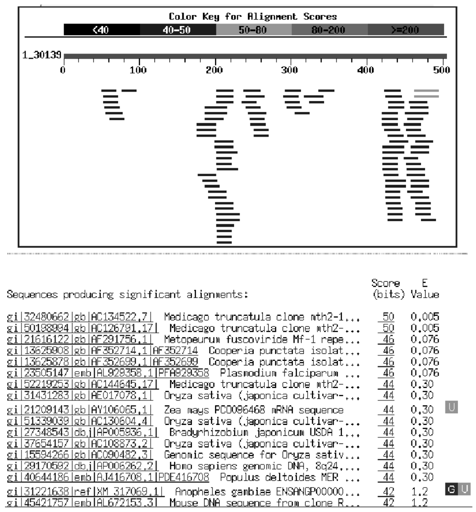

Figure 7: An example of BLASTN results with TTL system output (extended neighborhood selection: length 1→2). http://www.ncbi.nlm.nih.gov/BLAST/

for the "life". Such a system may prove to be advantageous in analysing evolutionary events in more detail.

- There are biases in the usage of truth tables similar to the codon bias in DNA data. These biases suggest that the TTL system does not merely output data at random. Thus, it is the meaningful experiments that we try to analyze these outputs, which have been generated randomly by initial settings.

Dealing with TTL systems as study subjects, we can evaluate the conventional techniques for analyzing molecular data. We may further propose new techniques for analyzing TTL systems, which can be used as feedback for bioinformatics.

References

[1] Fontana W. (1992). Algorithmic chemistry. In Langton CG, Taylor C, Farmar JD, Rasmussen S, editors, *Artificial Life II*, pages 159–209. Addison-Wesley, Redwood City, CA.

[2] Dittrich P, Ziegler J, Banzhaf W. (2001). Artificial chemistries — a review. *Artificial Life*, 7(3):225–275.

[3] Ray TS. (1992). An approach to the synthesis of life. In Langton CG, Taylor C, Farmar JD, Rasmussen S, editors, *Artificial Life II*, pages 371–408. Addison-Wesley, Redwood City, CA.

[4] Adami C. (1998). *Introduction to Artificial Life.* TELOS, Springer–Verlag, New York.

[5] Rasmussen S, Knudsen C, Feldberg R. (1992). Dynamics of programmable matter. In Langton CG, Taylor C, Farmar JD, Rasmussen S, editors, *Artificial Life II*, pages 211–254. Addison-Wesley, Redwood City, CA.

[6] Ikegami T , Hashimoto T. (1995). Active mutation in self-reproducing networks of machines and tapes. *Artificial Life*, 2(3):305–318.

[7] Banzhaf W. (1993). Self-replication sequence of binary numbers. foundation I and II: General and strings of length n=4. *Biological Cybernetics*, 69:269–281.

[8] Dittrich P, Banzhaf W. (1998). Self-evolution in a constructive binary string system. *Artificial Life*, 4:203–220.

[9] Eigen M, Schuster P. (1977). The hypercycle. a principle of natural self-organization. part a: Emergence of the hypercycle. *Naturwissenschaften*, 64:541–565.

[10] Barbieri M. (2003). *The Organic Codes : An Introduction to Semantic Biology.* Cambridge.

[11] Szathmáry E, Smith JM. (1997). From replicators to reproducers : the first major transitions leading to life. *Journal of Theoretical Biology*, 187:555–571.

[12] Gibas C, Jambeck P. (2001). *Developing Bioinformatics Computer Skills.* O'REILLY.