Historia Mathematica Vol. 12, No. 1, 1999, 73-81

셀룰러 오토마타 상에서 자기 복제

아주대학교 정보 및 컴퓨터공학부 **위규범**

Abstract

We survey the researches on self-reproducing structures on cellular automata. Self-reproduction is the foremost characteristic of life, and cellular automata are ideal model for studying artificial life. From the early studies by Von Neumann to late results on computational models using self-reproducing structures and emergence of self-replication are covered. Also possible applications of self-replicating structures are listed.

0. 서론

최근에 인공생명(Artificial Life)에 대한 관심이 높아지고 있다. 인공생명이란 생명체가 가지는 여러 특성들을 컴퓨터 모델링 또는 시뮬레이션을 이용하여 합성(synthesis)함으로서 생명 현상의 본질을 연구하는 분야를 일컫는다[1]. 인공적인 생명체를 합성하는 것 자체를 목표로 하는 연구자들도 있지만, 대개의 연구자들은 그 과정에서 얻어지는 생명 현상에 대한 이해를 목표로 한다. 현재 지구 상에 존재하는 생명체들 자체에 대한 연구가 아니라, 생명 현상의 일반적인 논리를 알고자 하는 것이다.

생명체의 여러 특성 중에서 자기 자신의 복제는 가장 기본적인 능력으로서, 인공생명의 분야에서도 중요한 연구 과제이다. 자기 복제 현상은 기계적인 장치 또는 컴퓨터 프로그램으로 구현할 수도 있지만, 본 논문에서는 셀룰러 오토마타 상에서의 자기 복제 현상에 대하여 논하고자 한다.

셀룰러 오토마타는 폰 노이만(Von Neumann)이 자기 복제 현상을 연구하기 위하여 제안한 모델로서[2], 이산적 동적체계(discrete dynamical system)이다. 이차원 격자(grid)를 이루는 셀(cell)들로 이루어지며, 각 셀은 각 시점(time step)에서 유한개의 가능한 상태 가운데 어떤 한 상태에 있으며, 다음 시점에 상태를 변화한다. 현재의 자신의 상태와 이웃하는 셀들의 상태에 의해서 다음 시점에서의 상태가 결정된다. 이렇게 상태가 변하는 규칙을 상태변

이함수(state transition function)이라고 부르며, 상태변이함수는 각 셀에 걸쳐서 동일하다. 이웃하는 셀들을 자신의 동, 서, 남, 북에 있는 셀들로 정의하는 경우(Von Neumann neighborhood)와 동, 서, 남, 북, 남동, 남서, 북동, 북서에 있는 셀들로 정의하는 경우(Moore neighborhood)가 있다.

1. 폰 노이만의 자기복제 장치

폰 노이만은 셀룰러 오토마타상에서 범용튜링기계(universal Turing machine)를 구현하고, 이것을 수정하여 universal constructor를 구현하였다. 범용튜링기계는 임의의 튜링기계 M과 임의의 문자열 w를 입력받아서, M이 w를 입력받아서 계산을 진행하는 과정을 시뮬레이트하는 기계이다. 튜링기계(Turing machine)는 양방향으로 무한히 긴 테이프에 문자열을 입력받아서, 입출력 헤드(head)가 테이프의 각 셀에 쓰여진 문자를 읽어서 다음 시점에 상태를 변화하고, 읽은 셀에 다른 문자를 쓰고, 헤드가 왼쪽 또는 오른쪽으로 한 셀 이동하는 장치로서, 일반적인 계산 모형이다. Universal constructor는 임의의 튜링기계 M을 테이프에 입력받아서(encoding된 형태로), M을 셀룰러 오토마타 상에 구축하고 자신의 테이프에 담겨 있는 내용을 M의 테이프에 그대로 복사하는 장치이다. 이제 universal constructor 자신의 인코딩을 자신의 테이프에 입력해주면, 이것이 바로 셀룰러 오토마타 상에서 자기 복제하는 장치임을 쉽게 알 수 있다(그림 1).

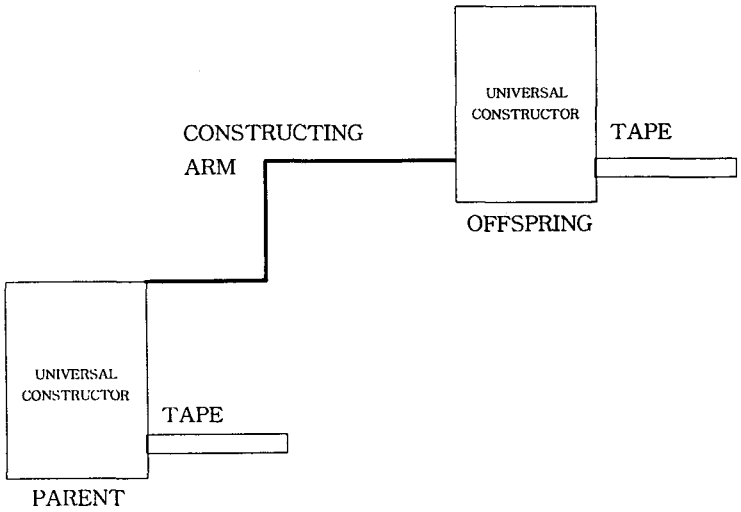


그림 1. 폰 노이만의 자기복제 장치

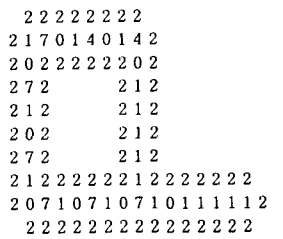
이렇게 구성된 폰 노이만의 자기복제 장치는 자체의 크기와, 각 셀이 29개의 상태를 가짐으로서 상태변화함수를 구현하는 테이블의 크기가 방대하여, 컴퓨터에서 구현해 보지 못하

였다. 폰 노이만의 자기복제 장치에서 한 가지 주의 깊게 볼 것은 universal constructor의 인코딩을 한 편으로는 해석하여(translate) 셀룰러 오토마타 상에 구현하면서, 다른 한 편으로는 universal constructor의 인코딩을 그대로 복사하여(transcript) 테이프에 담아 주었다는 것이다. 이렇게 해석과 복사가 동시에 일어나는 것은 생명체의 세포 분열 과정에서도 관찰된다.

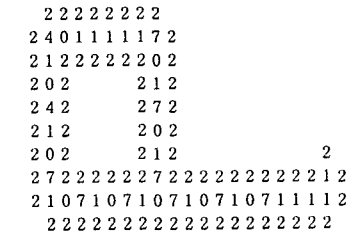
폰 노이만의 자기복제 장치는 그의 사후에 Arthur W. Burks에 의해서 세부사항들이 완성되고 발표되었다[3]. 폰 노이만의 장치는 워낙 방대하고 복잡하여, 이것을 단순화 시켜보려는 노력은 아주 자연스러운 일이었다. 1968년 E. F. Codd는 미시간 대학의 박사학위논문에서 각 셀의 상태가 8 개가 되도록 단순화하였으나, 여전히 universal constructor를 이용함으로서 컴퓨터에서 구현할 수 없을 정도로 크고 복잡하였다[4].

2. 랭튼의 자기복제 장치

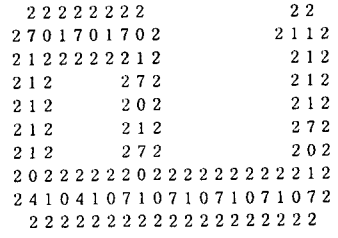
폰 노이만과 Codd의 자기복제 장치는 둘 다 universal constructor를 사용하였다. 여기서 한 가지 의문이 생긴다: "Universal computation은 자기복제의 필요 조건인가?" 여기에 대한 답은 1984년 랭튼(Chris Langton)에 의해서 주어진다[5]. 랭튼은 universal computation을 수행하지 못하는 아주 단순한 자기복제 구조를 셀룰러 오토마타에서 구현하였다(그림 2).



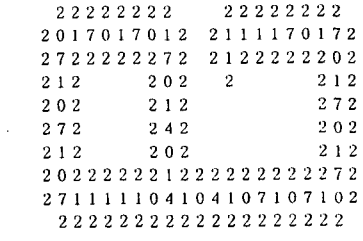
time 0



time 30



time 70



time 105

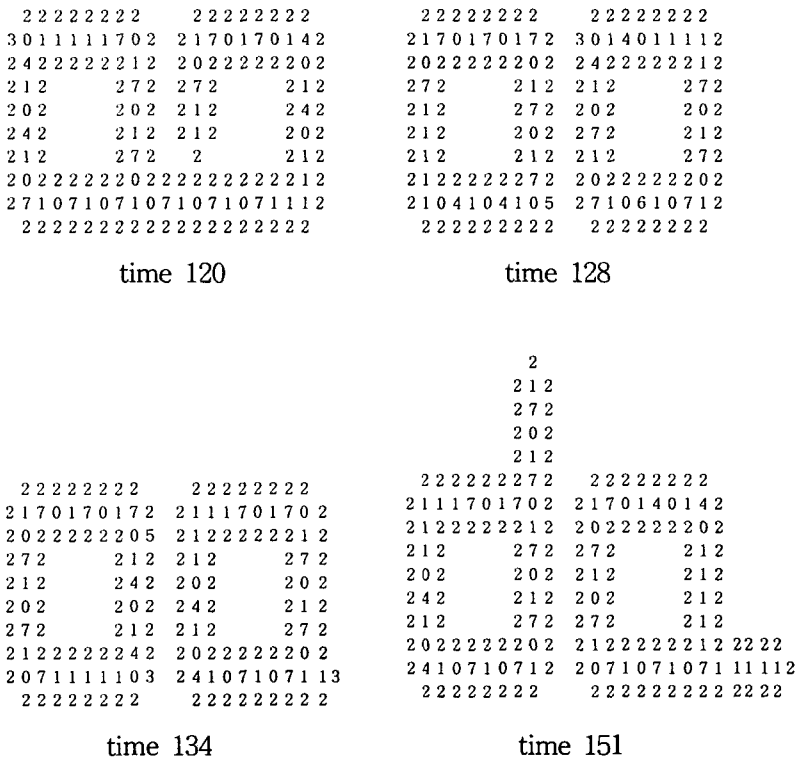
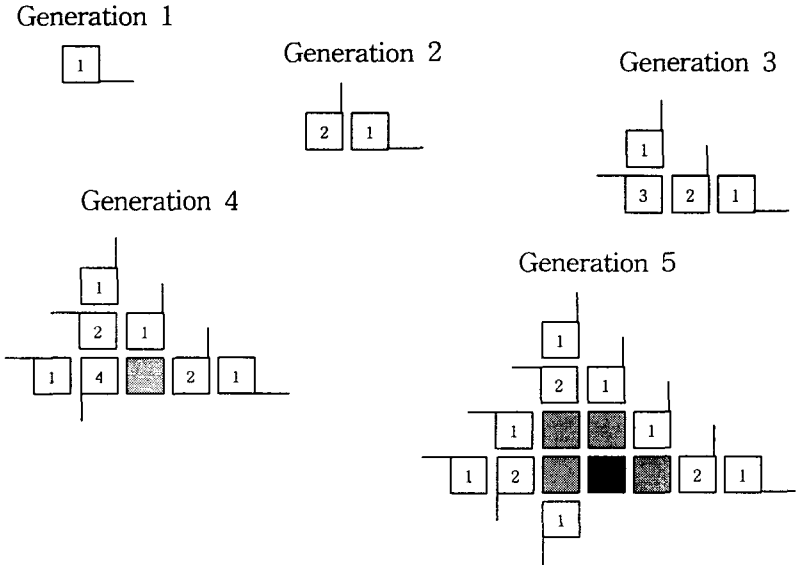


그림 2. 랭튼의 자기복제 구조

그림 2에서 보듯이, 151단위 시간 후에 자신과 똑같은 구조를 복제해낸다. 시간이 지나면서, 다수의 복제 구조가 생성되는 패턴은 그림 3과 같다.



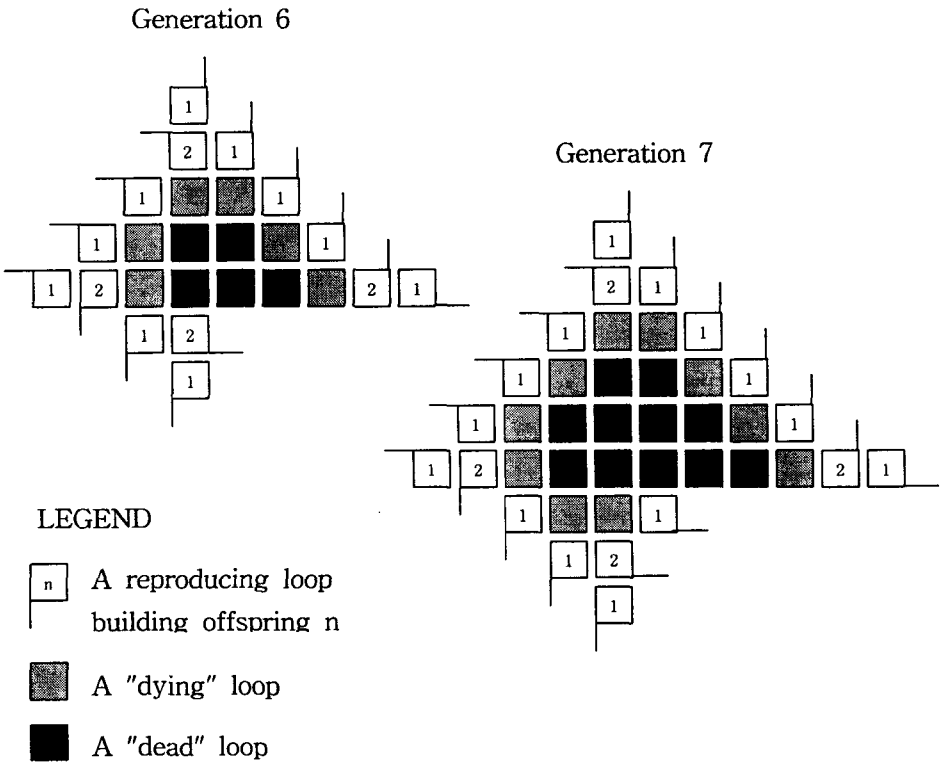


그림 3. 랭튼의 자기복제 구조들의 colony

랭튼의 자기복제 구조는 8개의 상태를 가지는 94개의 셀로 이루어지며, 상태변이함수를 저정하는 테이블의 크기도 214 밖에 되지 않는 단순한 구조이다. 랭튼의 구조에서도 해석과 복사가 동시에 진행됨을 볼 수 있다. 랭튼의 구조는 바깥을 싸고 있는 2 번 상태를 가진 셀들과 그 안에서 돌고 있는 시그널을 나타내는 내부 셀로 구별된다. 구조의 사각형 부분(loop)과 팔 부분(arm)이 접해 있는 부분에서는 내부에 돌고 있는 시그널이 팔 쪽으로 복사된다. 한 편 팔의 끝 부분에 도달한 시그널은 해석되어 자식 구조(daughter structure)를 구성하는(construct)역할을 한다. 예를 들어서 "7 0" 시그널은 직선 방향으로 진행하여 팔을 연장하도록 하고, "4 0" 시그널은 팔이 왼쪽으로 꺾여서 자라도록 한다.

랭튼의 자기 복제 구조는 universal computation을 필요로 하지 않는 획기적으로 단순한 구조로서 자기복제 연구의 전환점이 된다.

3. Byl 과 Reggia

랭튼의 자기 복제 구조는 Byl과 Reggia에 의해서 더욱 단순해진다. Byl은 랭튼의 구조에

서 외벽을 제거함으로서 더욱 단순한 구조를 만들었고[6], Reggia는 외벽과 내벽을 다 제거하는데 성공하였다[7]. Byl의 구조는 상태 5 개, 구조의 크기 12, 상태변이함수 테이블의 크기 66인 단순한 구조로서, 25 단위 시간 후에 초기의 자신의 모양과 똑같은 구조를 만들어낸다(그림 4).

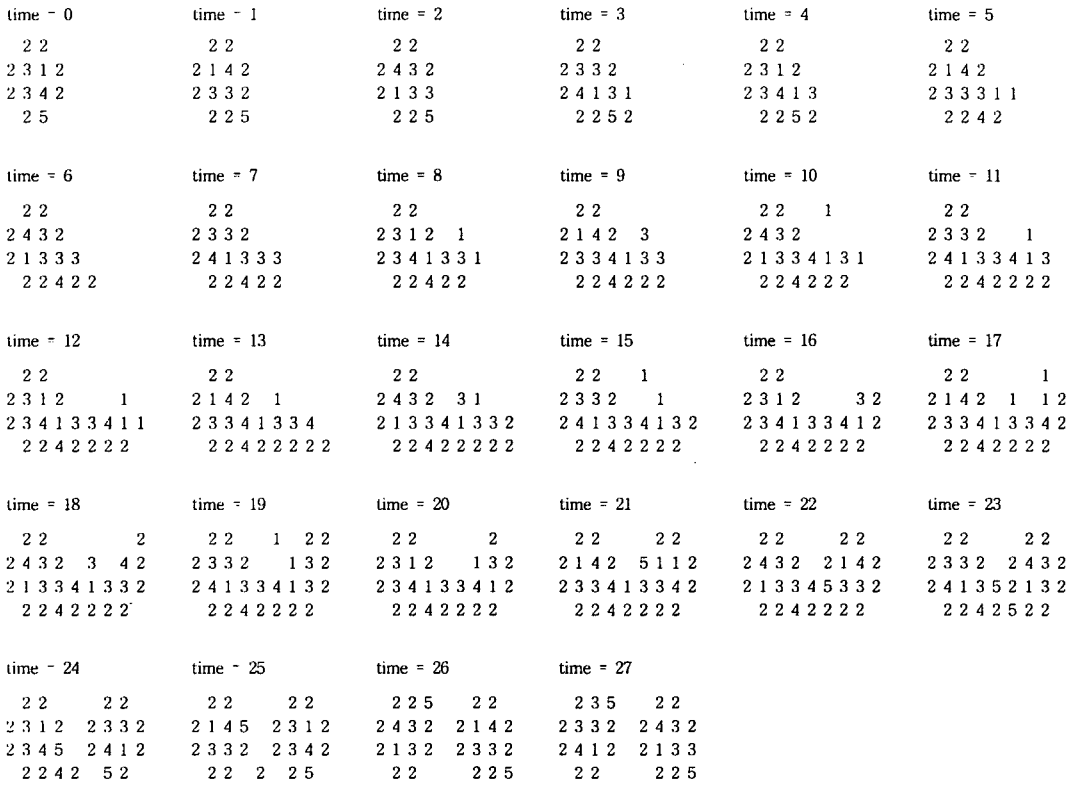


그림 4. Byl의 자기복제 구조

Reggia 등은 다양한 종류의 자기복제 구조를 만들었는데, 그 가운데 가장 크기가 작은 것은 상태 3 개, 크기 6, 상태변이함수 테이블의 크기 32, 복제 시간 14 인 단순한 구조를 가지고 있다.

4. 자기복제를 이용한 계산

자기복제하는 여러 가지 구조가 만들어지자, 이것을 이용하여 계산을 하고자 하는 시도를 하게 된다. 셀룰러 오토마타의 각 셀은 유한상태기계로서 아주 단순한 구조를 가지고 있고,

상태변이함수는 모든 셀에 걸쳐서 동일하므로, 셀룰러 오토마타는 효율적인 병렬 계산(parallel computation)의 모델로 적절하다. Tempesti[8]는 실행가능 프로그램(executable program)을 자기복제 구조에 덧붙여서, 일단 복제가 만들어지면 실행가능 프로그램도 따라서 복사가 되며, 이어서 프로그램이 실행된다. 시간이 지남에 따라 복제 구조의 수는 늘어나므로 동일한 계산을 동시에 여러 개의 프로세서로 진행하는 효과를 얻을 수 있다. 여기서의 문제점은 우리가 원하는 유용한 계산을 셀룰러 오토마타의 상태변이함수로 구현하는 어려움이다. 상태변이함수라는 단순한 국소적(local) 규칙을 적절히 디자인하여, 전역적(globally)으로 우리가 원하는 계산을 수행하도록 하는 것은 몹시 어려운 작업이다[9]. Tempesti가 덧붙인 프로그램은 자기가 속한 실험실의 로고인 LSL(logic systems laboratory, Swiss Federal Institute of Technology)를 자기복제 구조의 사각형 부분(loop)의 안쪽에 쓰게 만드는 단순한 프로그램이었다.

Chou[10]는 자기복제 구조를 이용하여 Satisfiablility 문제를 푸는 계산을 수행하였다. Satisfiability problem이란 아래의 식과 같은 *conjunctive normal form*의 *boolean* expression이 주어질 때, 이 boolean expression의 값이 true가 되도록 각 boolean variable에 true 또는 false를 assign하는 것이 가능한지 아니지 판정하는 decision problem이다.

$$(x_3 \vee \neg x_2) \wedge (\neg x_3 \vee x_1 \vee \neg x_4 \vee x_2) \wedge (x_4 \vee \neg x_1)$$

Satisfiability problem은 대표적인 NP-hard problem으로서 이 문제를 푸는 효율적인 알고리즘이 존재하지 않는다고 믿어지는 문제이다[11].

Chou는 단순한 자기복제 구조의 일부로서 boolean variable들의 true false assignment를 embed시키고, 각 셀에서 자기복제의 시그널들이 지나가는 것을 monitoring 하다가 그 복제 구조에 embed된 true false assignment가 주어진 boolean expression의 값을 false로 만드는 것이 확인되면, 그 자기복제 구조를 해체시켜(dissolve) 버린다. 해체되지 않은 자기복제 구조는 아직 true false 값이 assign되지 않은 다음 variable에, mother structure에서는 true를 assign하고 복제되는 daughter structure에서는 false로 assign한다. 이러한 과정을 반복하면 충분한 시간이 지난 뒤에는 주어진 boolean expression을 false로 만드는 assignment를 가진 구조는 모두 해체되어 없어지고, 주어진 boolean expression을 true로 만드는 assignment를 가진 자기복제 구조만 남아있게 되므로, 우리가 원하는 답을 얻을 수 있다.

Chou의 자기복제 구조는 mother와 완전히 똑같은 daughter를 만드는 게 아니라 1 bit 차이가 있도록 variation을 주는 것이다. Chou가 Satisfiability problem을 푸는 방법은 요즘 연구가 활발한 DNA computing[12]과 유사한 방식의 계산 모델로서, 효율적으로 구현할 수 있다면 NP-hard 문제들에 획기적인 해법이 될 수 있다.

5. 자기복제 구조의 창발 (Emergence of self-reproducing structures)

자기복제의 기능이 없는 단순한 구조들이 결합하여 자기복제 기능을 가지는 구조로 진화하는 과정은 생명의 기원을 밝히는 단서가 될 수 있는 흥미로운 문제이다. Chou 등은[13] 셀룰러 오토마타 상에 다양한 상태를 가지는 셀들을 무작위로 뿌려놓고 시간이 지남에 따라 이들이 결합하여 자기복제 구조로 진화하는 과정을 연구하였다. 이들의 실험에 의하면 81번의 실험 가운데 80번이나 자기복제 구조가 출현하였으며, 최초의 자기복제 구조는 4 개의 셀로 이루어지는 아주 작은 구조이며, 시간이 경과하면서 점점 더 큰 구조로 진화해 나감을 관찰할 수 있었다. 또한 자기복제 구조의 개수와 크기가 평형 상태에 도달하지 않고, 평균값 주위를 진동함으로서 혼돈역학계의 현상을 보임을 관찰하였다. 이 연구에서 사용한 상태변이함수는 기존에 자기복제하는 것으로 알려진 상태변이함수에 기반하여 디자인하였다.

Lohn 등[14]은 Chou와는 접근을 달리하여, 간단한 초기 구조를 주고 그 구조가 자기복제를 해나갈 수 있도록 하는 상태변이함수를 찾아가는 연구를 하였다. 이러한 상태변이함수를 찾아가는 방법은 유전알고리즘[15]을 이용하였다. Chou는 자기복제 규칙을 주고, 자기복제 구조가 생겨나는 과정을 연구하였고, Lohn은 구조를 주고 자기복제 규칙을 찾아나가는 방법을 제안한 것이다. 자기복제 구조와 규칙을 동시에 찾는 방법에 대한 연구는 무척 흥미로울 것이다.

6. 결론

셀룰러 오토마타 상에서 자기복제 구조에 대하여 살펴 보았다. 이들은 mathematical biology의 입장에서, 또한 computational paradigm 으로서 매우 흥미있는 문제들이다. 또한 이러한 연구들은 자기복제하는 분자구조를 합성하거나, nanotechnology에 이용될 수 있으며, 전자회로 자체가 자기복제하여 유용한 계산을 수행하도록 하는 등 다양하고 유망한 응용의 가능성을 가지고 있다([16], [17]).

참고 문헌

1. C. G. Langton, *Artificial Life: An Overview*, MIT Press, 1996.
2. J. von Neumann, *The Theory of Self-Reproducing Automata*, University of Illinois Press, 1966.
3. A. Burks, "Von Neumann's Self-reproducing Automata," *Essays on Cellular Automata*, ed. A. Burks, University of Illinois Press, 1970, 3-64.
4. E. Codd, *Cellular Automata*, Academic Press, 1968.

5. C. G. Langton, "Self-Reproduction in Cellular Automata," *Physica D* 10(1984), 135-144.
6. J. Byl, "Self-Reproduction in Small Cellular Automata," *Physica D* 34(1989), 295-299.
7. J. Reggia, S. Armentrout, H.-H. Chou, and Y. Peng, "Simple Systems that Exhibit Self-Directed Replication," *Science* 259(1993), 1282-1287.
8. G. Tempesti, "A New Self-Reproducing Cellular Automaton Capable of Construction and Computation," *Lecture Notes in Artificial Intelligence* 929(1995), 555-563.
9. M. Mitchell, J. Crutchfield, and P.Hraber, "Evolving Cellular Automata to Perform Computations: Mechani는 and Impediments," *Physica D* 75(1994), 361-391.
10. H.-H. Chou and J. Reggia, "Problem solving during artificial selection of self- replicating loops", *Physica D* 115(1998), 293-312.
11. M. Gary and D. Johnson, *Computers and Intractability*, Freeman, 1979.
12. L. Adleman, "Molecular Computation of Solutions to Combinatorial Problems," *Science* 266(1994), 1021-1024.
13. H.-H. Chou and J. Reggia, "Emergence of self-replicating structures in a cellular automata space," *Physica D* 110(1997), 252-276.
14. J. Lohn and J. Reggia, "Automatic Discovery of Self-Replicating Structures in Cellular Automata," *IEEE Transactions. on Evolutionary Computation* 1(1997), 165-178.
15. M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, 1996.
16. K. Drexler, *Nanosystems: Molecular Machinery, Manufacturing, and Computation*, John Wiley, 1992.
17. D. Mange, M. Goeke, M. Madon, D. Stauffer, A. Tempesti, and S. Durand, "Embryonics: A New Family of Coarse-Grained Field-Programmable Gate Arrays with Self-repair and Self-reproducing Properties," *Towards Evovable Hardware*, ed. E. Sanchez and M. Tomassini, Springer, 1996, 197-220.