1차원 셀룰러 오토마타에 의한 진화시스템의 설계 및 구현

박 정희
양산전문대학 전자계산과

A Design and Implementation of the Evolution System by one dimensional Cellular Automata

Jung-Hee,Park
Dept. of Computing Science
YangSan College

Abstact

Cellular Automata are a discrete mathematical system whose evolution is governed by a deterministic rule involving local interactions. In this paper, one designed and implemented the evolution system based on GUI which can analyse how random initial states evolve easily.

I. Introduction

Cellular automata are discrete dynamical systems, of simple construction but complex and varied behaviour. First introduced in 1948 by Von Neumann and Ulam[1] as potential models for biological self-reproduction, cellular automata have since been used as mathematical tools for studying a wide variety of problems. In general, cellular automata can be defined as a spatial lattice of sites whose values at each time step t are determined as a transition function of the

values of neighboring sites at the previous time step t-1 [2]. This function provides the rule governing the automata's behavior. Specially, let us consider the class of automata defined on one dimensional set of sites x_i, each of which assumes any of the values V={0,1} (k=2). The general form of a rule for such an automaton is then given by

$$x_i^{t+1} = f(x_{i-r}^t, \ldots, x_i^t, \ldots, x_{i+r}^t) ,$$
$$f : \{0,1\}^{2r+1} \rightarrow \{0,1\}$$

where $r \geq 0$ represents the size of the neighborhood considered by the rule and each site x_i is assigned an initial value x_i^0.

Elementary cellular automata of r=1 are defined by rules of the form:

$$x_i^{t+1} = f(x_{i-1}^t, x_i^t, x_{i+1}^t) ,$$
$$f : \{0,1\}^3 \rightarrow \{0,1\}$$

A rule is therefore equivalently defined by specifying the value assigned to each of the 2^3 possible 3-tuple configurations of site values; i.e., by specifying the a_i , i=0,...,7, such that

111	110	101	100	011	010	001	000
↓	↓	↓	↓	↓	↓	↓	↓
a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0

Since each $a_i \in \{0,1\}$, there is a total 256 possible elementary rules. Wolfram has provided an extensive catalog of the behavior associated with different choices of the rules combined with different initial conditions. In particular, on the basis of systematic computer simulation of a large number of automata, he conjectures that all automata belong to at least one of four classes, qualitatively charaterized as follows[3,4,5]:

Class 1: A fixed, homogeneous state is evetually reached(e.g. rules 0,8,128,136,250)

Class 2: A pattern consisting of seperated periodic regions is produced (e.g. rules 4,6,12,20,37,56,72, 73,108,178)

Class 3: A chaotic, aperiodic pattern is produced(e.g. rules 18,45,90,126,146)

Class 4: Complex, localized structures are generated

In this paper, one designed and implemented of the evolution system for k=2, r=1 cellular automata[2] with boundary condition 0-0.

An invariant string of an automaton rule is defined to be a finite spatial sequence of site values that remains invariant under the rule, independent of the sequence's spatial position or the values of its neighboring sites. In other words, invariant strings will be preserved by the automaton and any other string, representing "noise" in the input will be annihilated. The evolution process of the some rules for each Wolfram's class from disordered states will be shown in section 2.

II. The evolution system

This section shows patterns produced by evolution for some rules of Wolfram's each class. starting from random initial conditions. For the implementation of this program, the IBM-PC AT 486 DX2 and Turbo C programming language were used. This system has 30 * 30 cells and the first row is the random initial states which user clicked the mouse randomly. The marked cells represent 1, the others represent 0. Clicking the "evolution" button at right top, the system represents the evolution process automatically from the second row. Fig.1 shows the evolution system of CA-8 in class 1. The states by CA-8 converge to all 0's. Fig.2 shows that of CA-6 in class 2. The states by CA-6 take longer time step to become stable than that by CA-8. Fig.3 shows that of CA-18 in class 3. We can see that the evolution states by CA-18 are chaotic.

Fig.1 rule 8 (00001000)

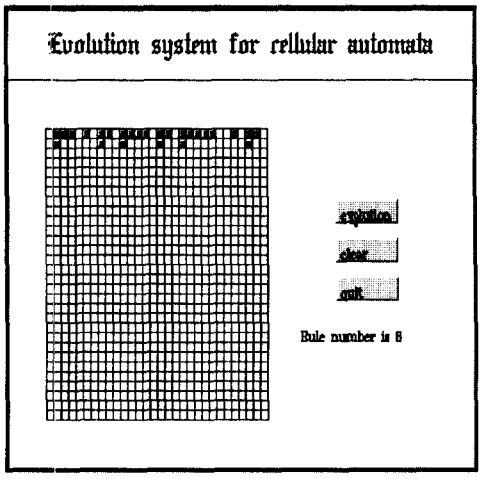


Fig.2 rule 6 (00000110)

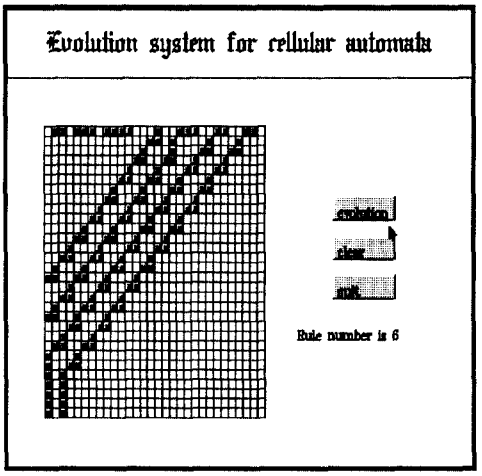
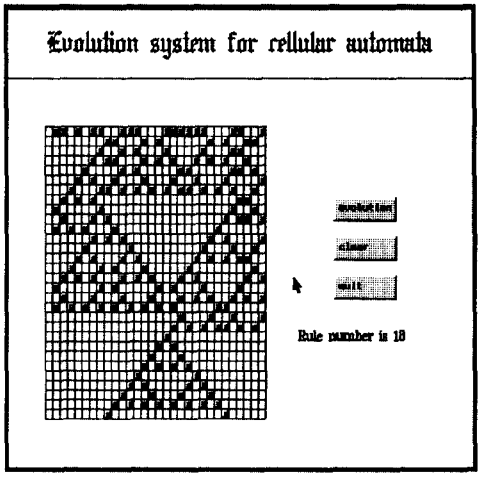


Fig.3 rule 18 (00010010)



III. Conclusion

In this paper, we designed and implemented the evolution system based on GUI by one dimensional cellular automata. This can help us to

easily understand how one dimensional cellular automata evolve. However since one used Turbo C programming language for the implementation, this makes the program length long and the programming is rather cumbersome. The use of window programming language such as Visual Basic will give better GUI. Therefore the conversion this system to that for window programming language is recommended.

References

1. J. Von Neumann, "Theory of self-reproducing automata", A.W. Burks,ed.

2. Erica Jen, "Invariant strings and pattern-recognizing properties of one-dimensional cellular automata", Journal of statistical physics, Vol.43, Nos. 1/2, 1986

3. Erica Jen, "Global properties of cellular automata", Journal of statistical physics, vol.43, nos. 1/2, 1986.

4. S.Wolfram, "Universality and complexity in cellular automata", physica D,volume 10,pages 1-35, January,1984.

5. S.Wolfram , "Cellular automata and complexity:collected paper", Addison-wesley, 1994

6. S.Wolfram, "Twenty problems in the theory of cellular automata", proceedings of the fifty-ninth Nobel symposium, 1985

7. Lee, Hyen Yeal, "Studies on Dynamical Behaviors of Finite Cellular Automata", Ph.D Thesis,1995

8. G. Weisbuch ,"Complex Systems Dynamics", Addison-wesley,1991

9. T. Toffoli, N. Margolus, "Cellular Automata Machines", The MIT Press Cambridge, Massachusetts London, England