Discovery of Cellular Automata Rules Using Cases

Ken-ichi Maeda

Chiaki Sakama

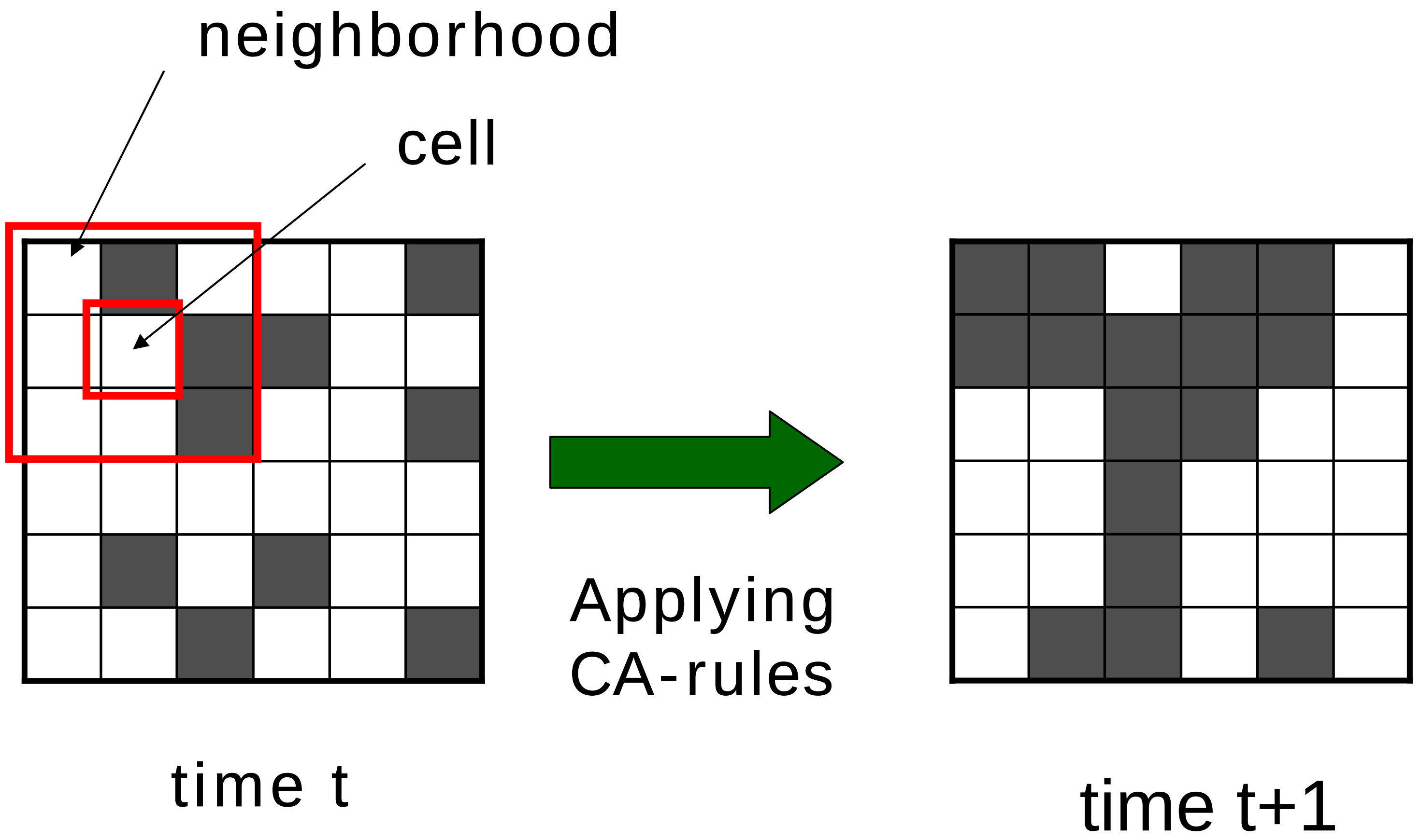
Wakayama University

Discovery Science 2003, Oct.17

Cellular Automata (CA)

- A CA consists of a regular grid of cells.
- A cell has a finite number of possible states.
- The state of each cell changes synchronously in discrete time steps according to local and identical transition rules (CA-rules).
- The state of a cell in the next time step is determined by its current state and the states of its surrounding cells (neighborhood).

Example: 2-dimensional 2-state CA

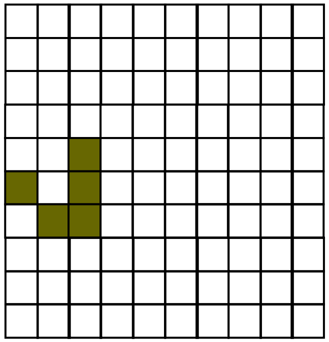


Game of Life (LIFE)

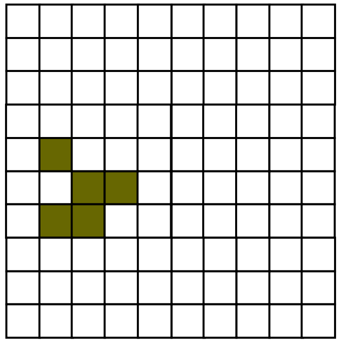
- The state of a cell is either “black or white”.
- A neighborhood consists of a central cell and the nearest 8 surrounding cells.
- Cellular states change according to the CA-rules:

1. If the central cell has exactly 2 surrounding black cells, the next state of the cell does not change.
2. Else if the central cell has exactly 3 surrounding black cells, the next state of the cell is black.
3. Otherwise, the next state of the central cell is white.

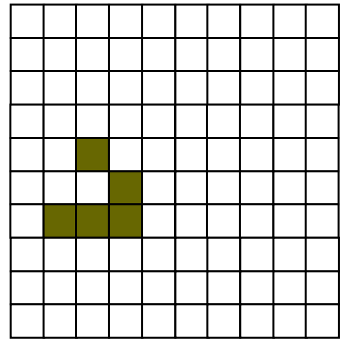
The “Glider” Pattern in LIFE



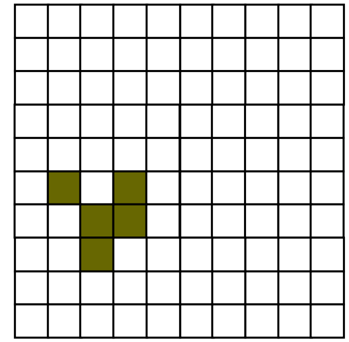
S_0



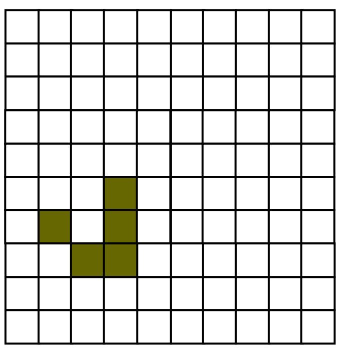
S_1



S_2



S_3



S_4

……

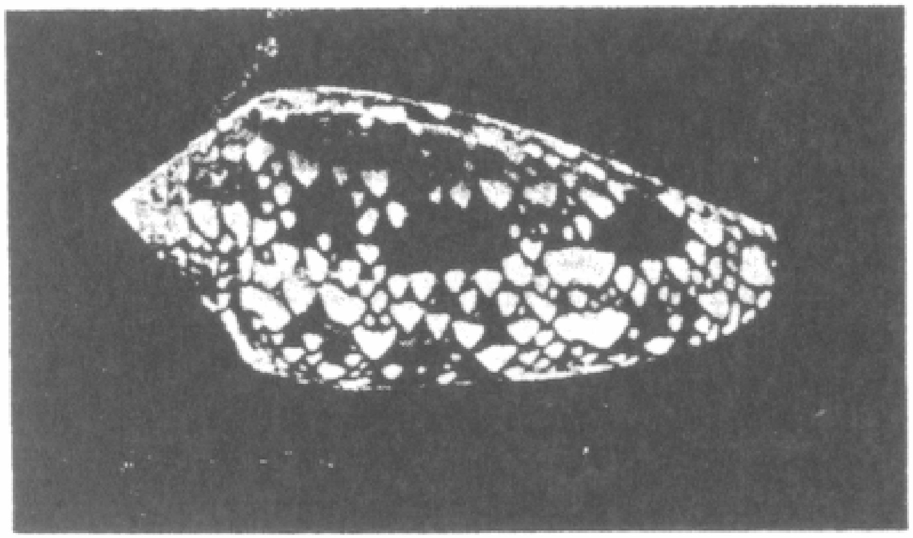
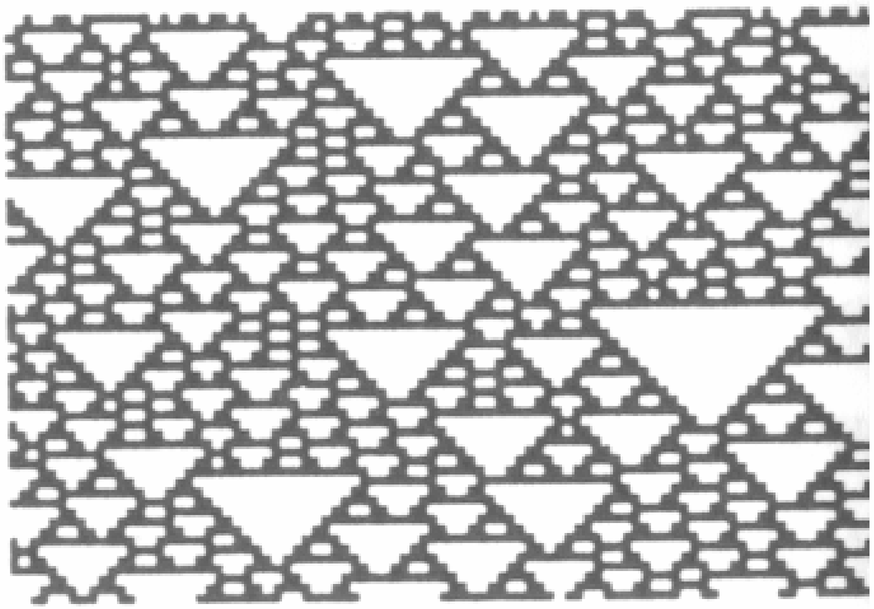
★ LIFE emerges complex behavior as a self-organizing system.

Applications of CAs

- CAs are used for modeling advanced computation such as massively parallel computers and evolutionary computation.
- CAs are used for simulating various complex systems in the real world including biological, chemical, physical and sociological systems.

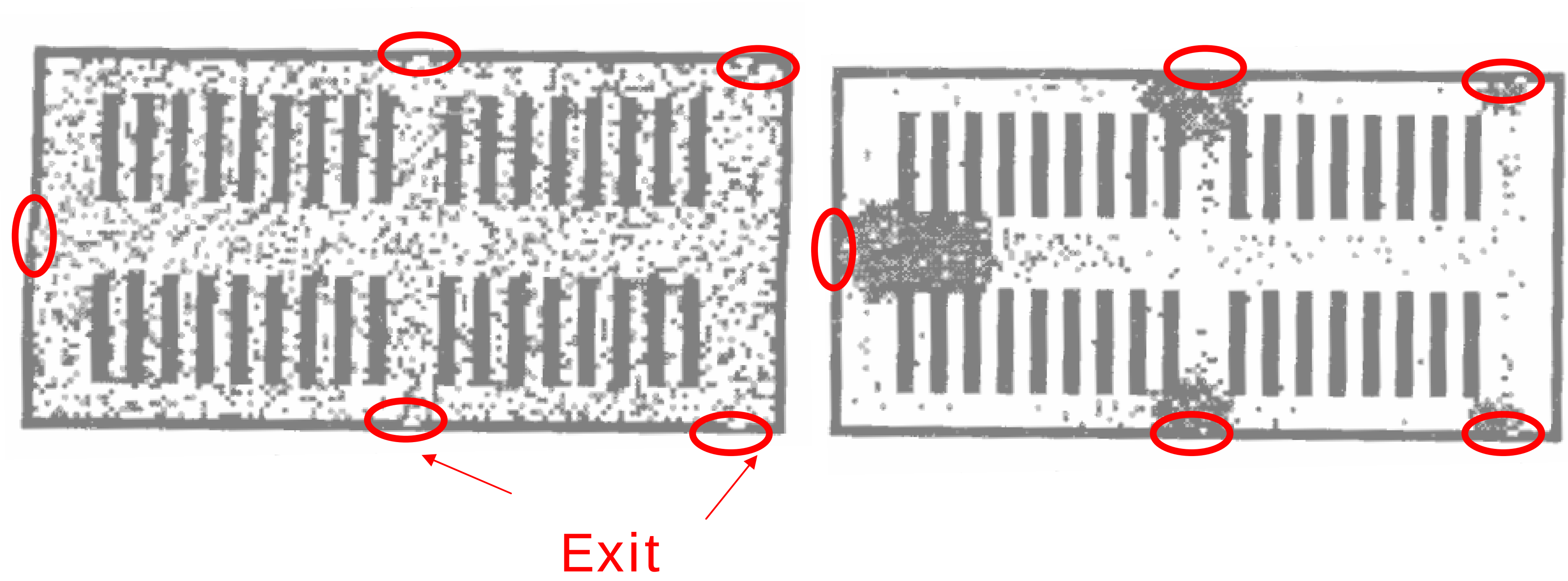
Example: Simulation of Self-Organizing Systems

- Geometric patterns appearing in nature.

Example: Simulation of behaviors of animals and humans

- People escape by the exit.



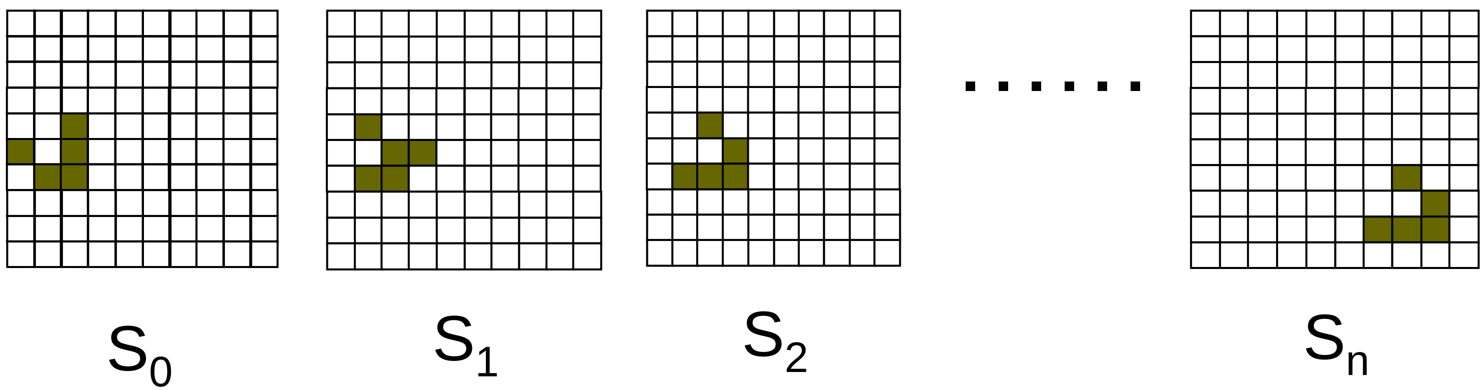
Purpose of this Research

- Complex behavior of CAs is difficult to understand, which makes hard to design CAs having desired behavior.
- The task of designing CAs is done by human experts and it becomes harder as a target problem becomes complicated.

⇨ The purpose of this research is to develop a method for automatic discovery of CA-rules to support the task of designing CAs.

Problem Setting

Given: a sequence of CA configurations



Find: a set of CA-rules which produce the change of patterns appearing in the input configurations.

Proposed Method

1. From input configurations, determine an appropriate neighborhood using hill-climbing search.
2. Collect cellular changes of states as cases from input configurations. A case is defined as a pair of ‘a neighborhood’ and ‘the next state’.
3. Construct a decision tree to classify cases. Conditions for classifying cases in a decision tree are computed using genetic programming.
4. Extract CA-rules from a decision tree. Each CA-rule has the form: ‘if (the current state of a cell and its neighborhood), then (the next state of the cell)’.

Experimental Results

- Given 2-dimensional 2-state CA configurations produced by a 2-dim. 2-state CA, the algorithm found the original 2-dim. 2-state CA-rules which reproduce the same configurations.
  (LIFE is used in this experiment.)
- Given 2-dimensional 2-state CA configurations produced by a 1-dim. 2-state CA, the algorithm discovered new 2-dim. 2-state CA-rules which reproduce the same configurations.

Conclusion and Future Work

- We developed an algorithm for automatic generation of cellular automata rules.
- We verified by experiments that the proposed algorithm successfully finds CA-rules which reproduce input configurations.
- In real-life problems input configuration generally include noise, then it is necessary to have a mechanism of classifying cases in face of noise.