Cellular Automata I

Modelling Complex Systems

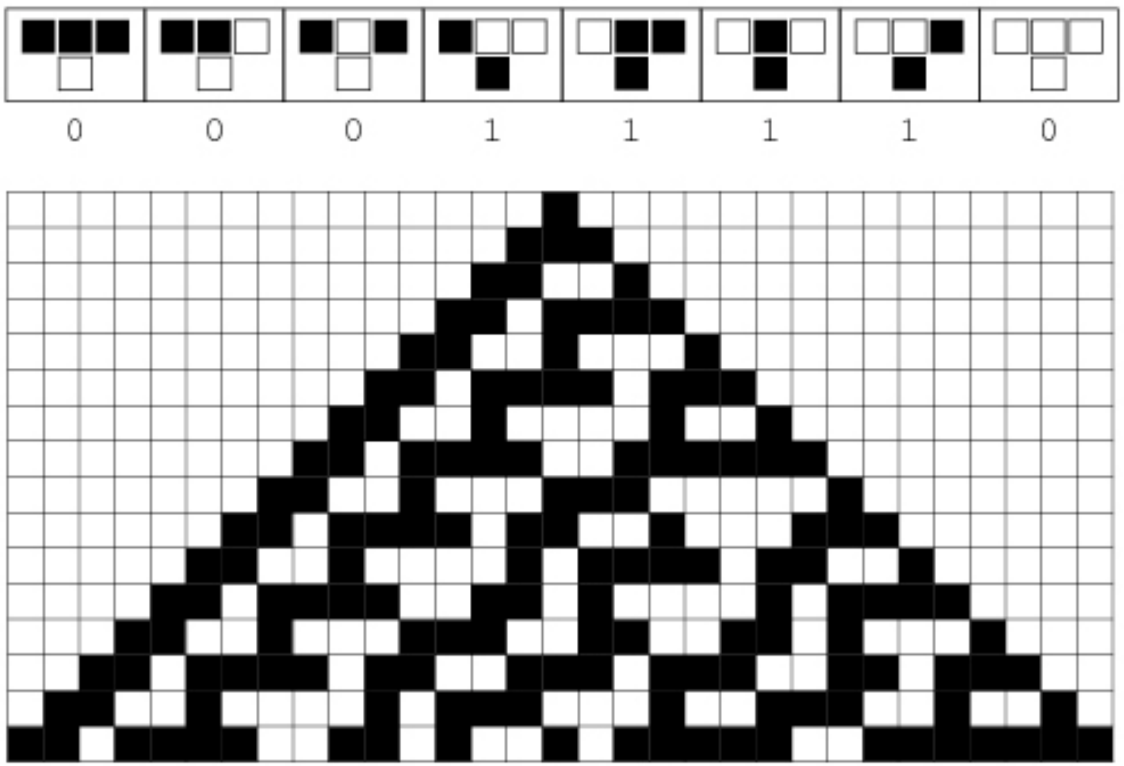
What is a cellular automata (CA)?

A CA consists of an array of cells each with an integer “state”.

On each time step a local update rule is applied to the cells. The update rule defines how the particular cell will update its state as a function of its neighbours state.

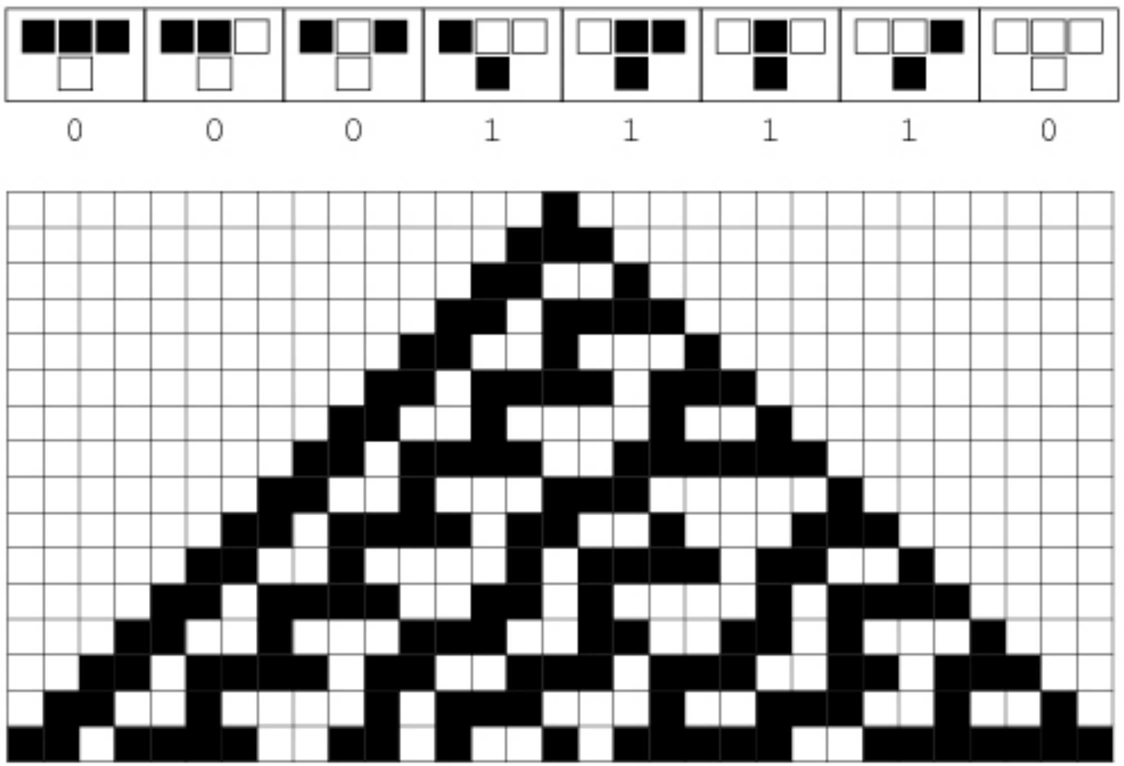
The CA is run over time and the evolution of the state is observed.

elementary cA



- white = 0, black = 1
- 111 —> 0
  110 —> 0
  101 —> 0
  100 —> 1
  011 —> 1
  010 —> 1
  001 —> 1
  000 —> 0

elementary cA



- 2^8 = 256 rules in total
- rule 0 — rule 255

elementary cA

- Classified based on patterns
- Class 1: **Fixed**; all cells converge to a constant 0 or 1 set

  Class 2: **Periodic**; repeats the same pattern, like a loop

  Class 3: **Chaotic**; pseudo-random

  Class 4: **Complex local structures**; exhibits behaviours of both class 2 and class 3; with long lived hard to classify structure
- Feels that we understand class 1 - 3, but not 4.

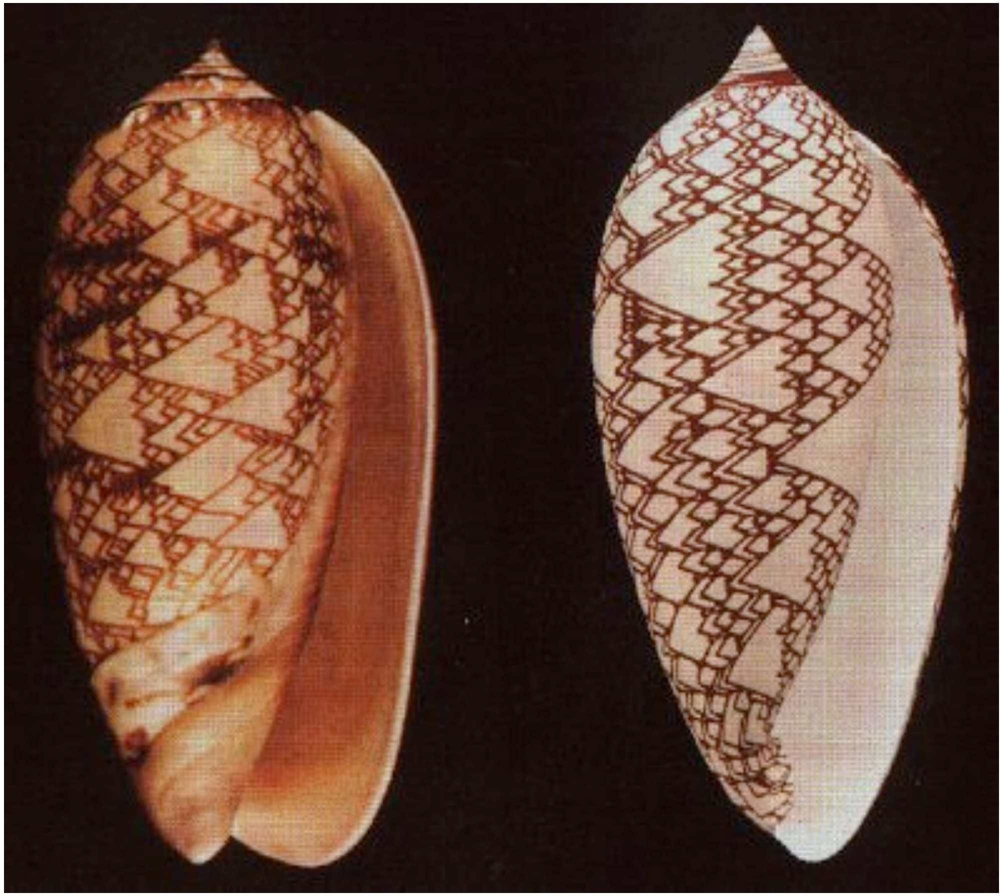
elementary cA

- Class 1: **Fixed**; e.g., rule 8 (00001000)
- Class 2: **Periodic**; e.g., rule 50 (00110010)
- Class 3: **Chaotic**; e.g., rule 30 (00011110)
- Class 4: **Complex local structures**; e.g., rule 110 (01101110)

elementary cA

- Class 1: **Fixed**; e.g., rule 8 (00001000)
- Class 2: **Periodic**; e.g., rule 50 (00110010)
- Class 3: **Chaotic**; e.g., rule 30 (00011110)
- Class 4: **Complex local structures**; e.g., rule 110 (01101110)

elementary cA



More complex cA

- CA can be extended:

1. More states for single grid
2. Longer range interactions
3. Two or more dimensions
4. Hexagonal or other grids

……

Formal definition

- We start from a **configuration**.
- All the cells update **simultaneously** their colour, and choose their new colour in function of the colours they observe in a **finite neighbourhood**.

If all cells apply simultaneously the same local rule, the update dynamics is called a **cellular automaton**.

Formal definition: cellular automata on inf line

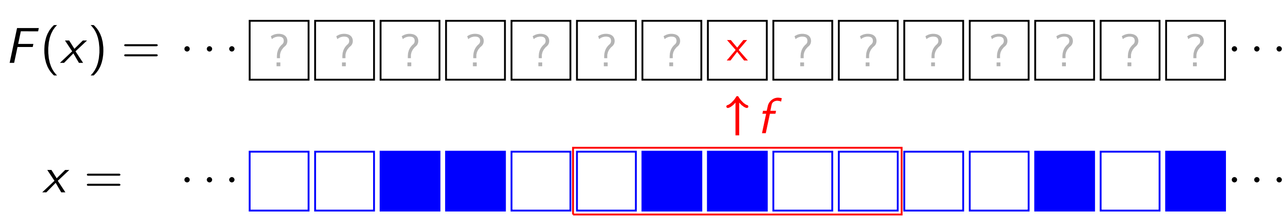
Let $\mathcal{A}$ be a finite set of symbols, called the **alphabet**.

We denote by $\mathcal{A}^{\mathbb{Z}}$ the set of **configurations**.
An element of $\mathcal{A}^{\mathbb{Z}}$ is a sequence $(x_k)_{k\in\mathbb{Z}}$, with $x_k \in \mathcal{A}$ for $k \in \mathbb{Z}$.

Definition

A map $F : \mathcal{A}^{\mathbb{Z}} \to \mathcal{A}^{\mathbb{Z}}$ is a **cellular automaton** if there exists a **radius** $r \geq 0$ and a **local function** $f : \mathcal{A}^{2r+1} \to \mathcal{A}$ such that:

$$F(x)_k = f(x_{k-r}, \dots, x_{k+r-1}, x_{k+r}).$$



$\mathcal{A} = \{\square, \blacksquare\}$, $r = 2$

Irene Marcovici

Game of life

- **World**: 2D orthogonal grid of square cells
- **States**: Dead (0, white) or Alive (1, black)
  - Reproduction: 0 -> 1, if #(Alive neighbours) = 3
  - Surviving: 1 -> 1, if #(Alive neighbours) = 2 or 3
  - Underpopulation: 1 -> 0, if #(Alive neighbours) < 2
  - Overpopulation: 1 -> 0, if #(Alive neighbours) > 3
  - Otherwise, no change

Game of life

- Reproduction:

0 -> 1, if #(Alive neighbours) = 3

- Surviving:

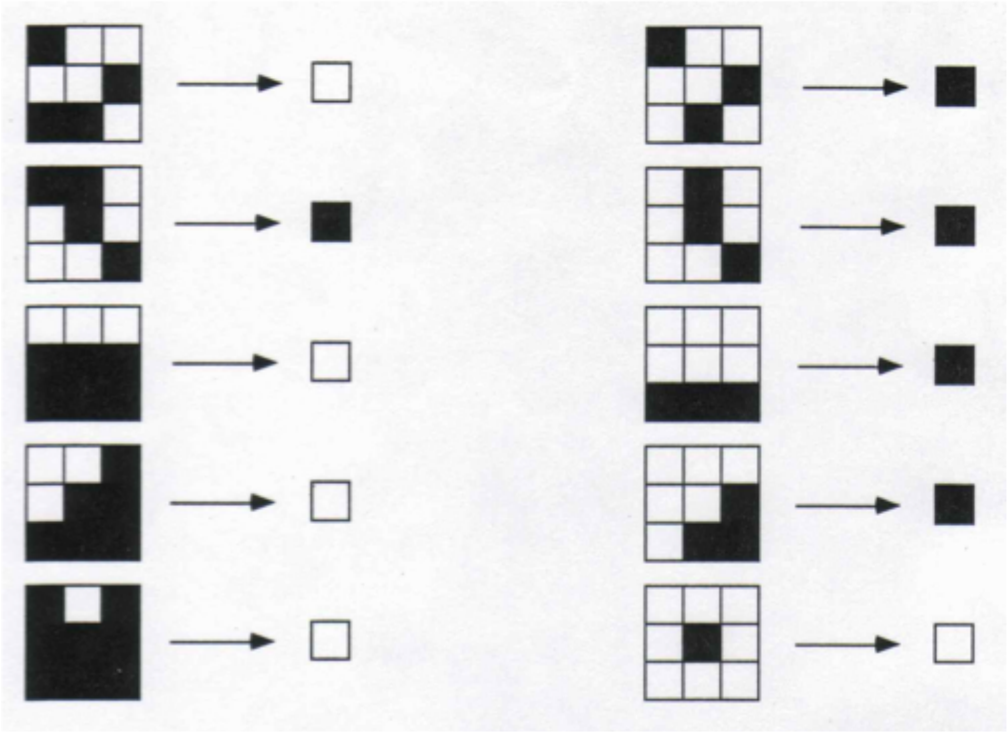
1 -> 1, if #(Alive neighbours) = 2 or 3

- Underpopulation:

1 -> 0, if #(Alive neighbours) < 2

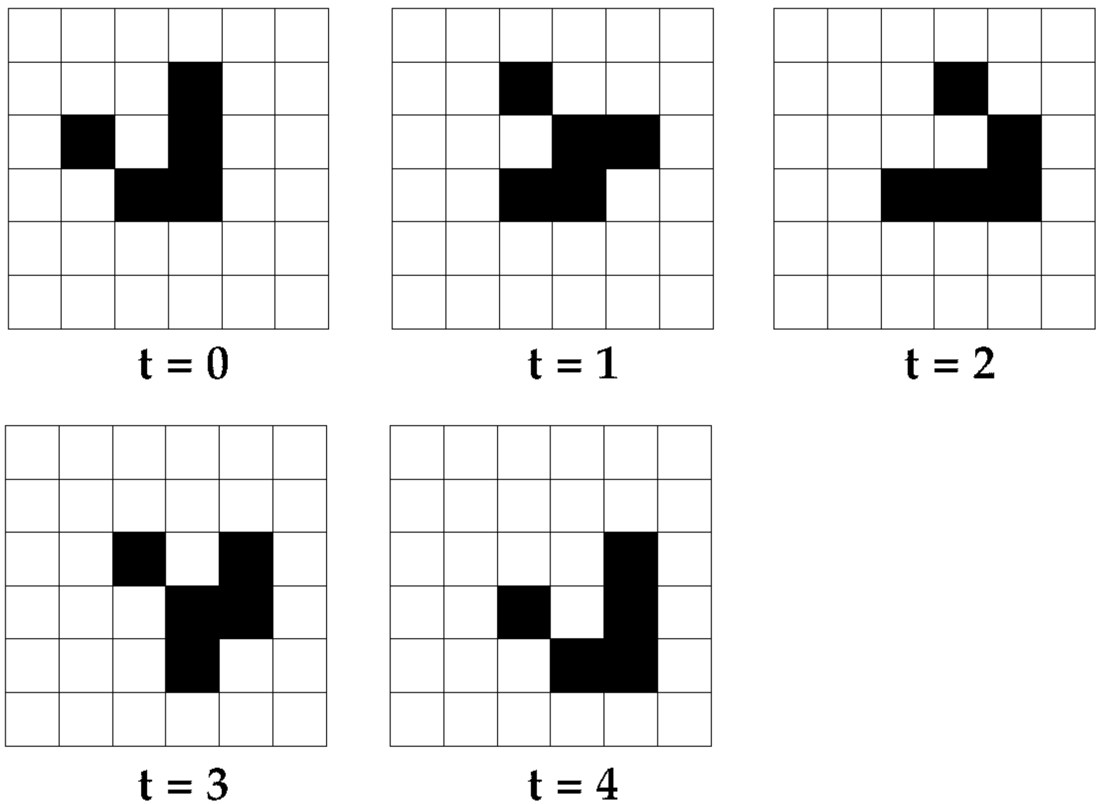
- Overpopulation:

1 -> 0, if #(Alive neighbours) > 3



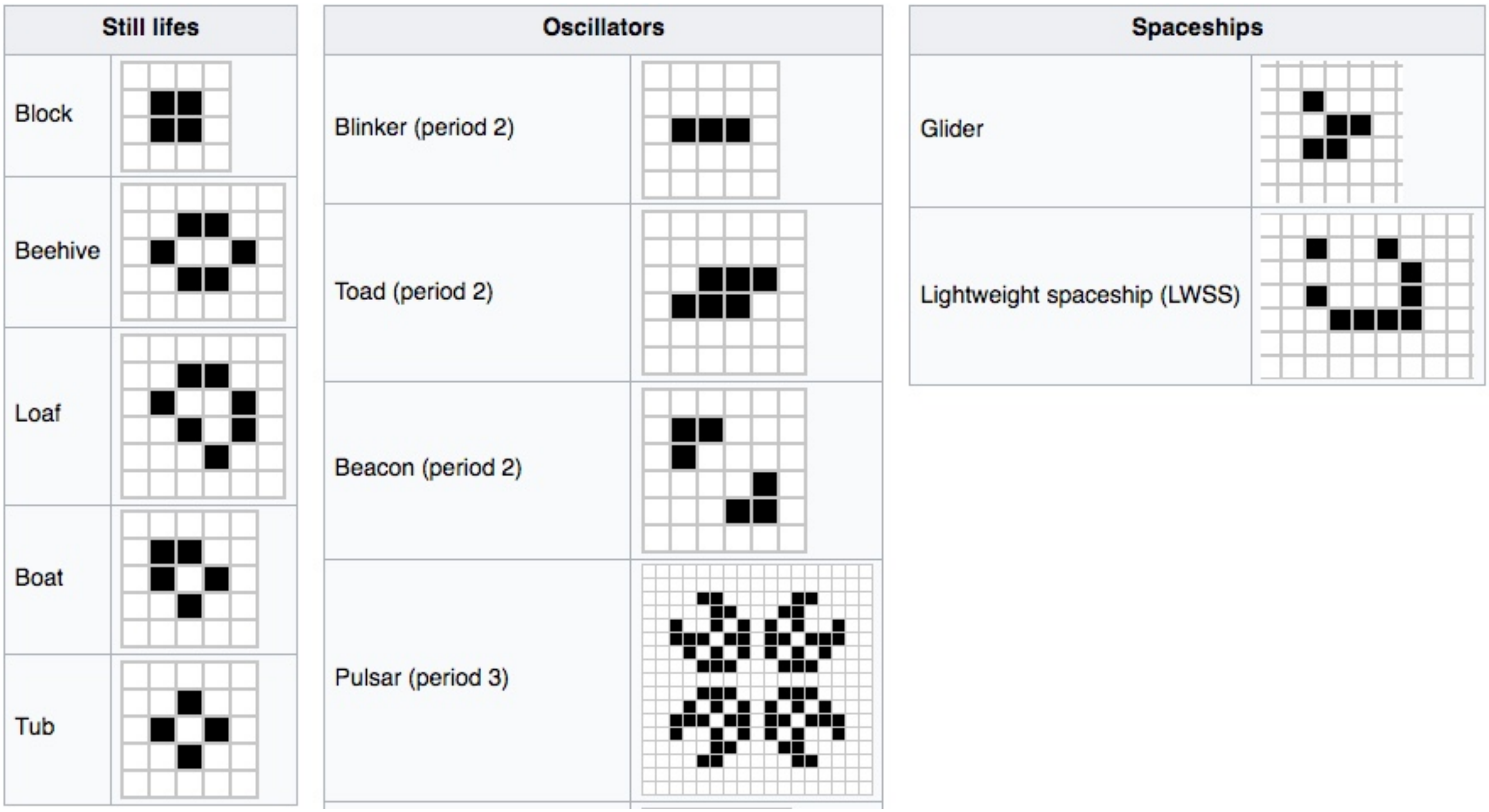
Game of life

- Example: Glider



Game of life

- More examples:



https://en.wikipedia.org/wiki/Conway's_Game_

Large-scale structures

https://vimeo.com/5428232

Computational gates

https://www.youtube.com/watch?v=vGWGeund3eA

Bellos (2015) Alex Through The Looking Glass