Models of Computation

11: Cellular Automata

Cellular Automata

- A cellular automaton is a parallel model of computation. Intruduced by John Neumann and Stanislaw Ulam in 1940's. Properties:
  - Automata, called “cells”, that are interconnected.
  - Any two cells having the same local neighborhood.
  - The cells are uniform and simple, their only characteristic is their state.
  - Each cell sees only its neighboring cells and updates its state based on the state of neighboring cells and its own state.
  - The cells operate synchronously in discrete time steps.

Cellular Automata

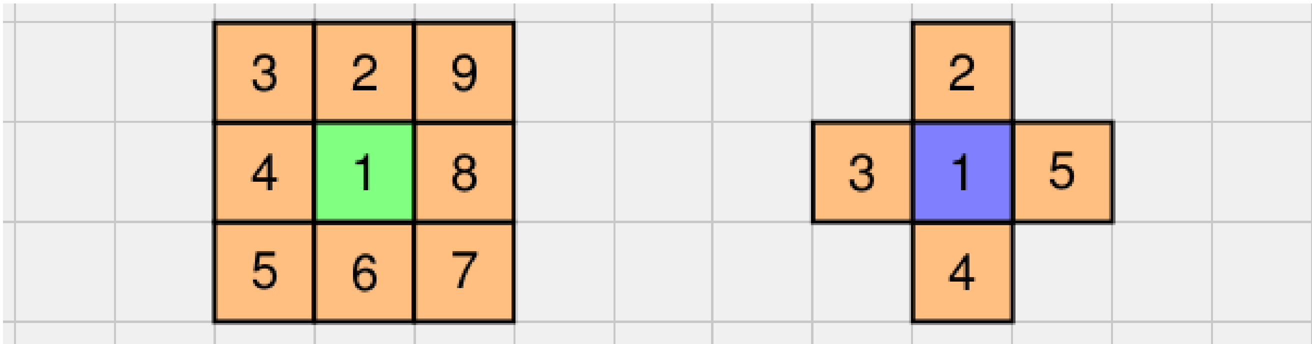
- A **cellular automaton** A = <X , S , N , f> is a 4-tuple:
  - X is an infinite subset of a vector space, called **cellular space**. The elements of X are called **cells**.
  - S is a non empty finite set of **cell states**.
  - N = (n_1 , . . . , n_m) a vector of m elements, the **neighborhood vector**, s.t. $x + n_i \in X$, for $\forall x \in X$, $1 \le i \le m$.
  - $f : S^m \to S$ is the **local updating rule**.

- If the new state of a cell also depends on its current state, then $0 \in N$.
- If $X = Z^d$ and $n_i \in Z^d$ ($1 \le i \le m$) then the condition for the neighborhood vector is satisfied. Then we can briefly write <d , S , N , f>.
- We mostly deal with the case $X = Z^d$, but other spaces are also possible, e.g. a hexagonal or triangular grid on the Euclidean plane, or torus or hyperbolic plane.

Cellular Automata

- Let $N = (n_1, \ldots, n_m)$. The **set of neighbors of a cell** $x \in X$ is $N(x) = \{x+n_i \mid 1 \leq i \leq m\}$.
- A mapping $c : X \rightarrow S$ is called a **configuration**.
- Let $A = <X, S, N, f>$ be a cellular automaton, where $N = (n_1, \ldots, n_m)$ and $c : X \rightarrow S$ is a configuration. Then we define $G : S^X \rightarrow S^X$ the **global state transition function**.
  Let $x \in X$ an arbitrary cell,
  $$G(c)(x) := f(c(x+n_1), \ldots, c(x+n_m)).$$

Neumann- and Moore-Neighborhood



Moore Neumann

- If $X = Z^2$, the Moore-neighborhood of the green cell is 2-9, the Neumann-neighborhood of the blue cell is 2-5.
- If we want the new state of a cell to depend on its own previous state in addition to the states of its neighboring cells, then
  - $N_{Moore} = ((0,0), (0,1), (-1,1), (-1,0), (-1,-1), (0,-1), (1,-1), (1,0), (1,1))$.
  - $N_{Neumann} = ((0,0), (0,1), (-1,0), (0,-1), (1,0))$.

Game of Life (Conway 1970)

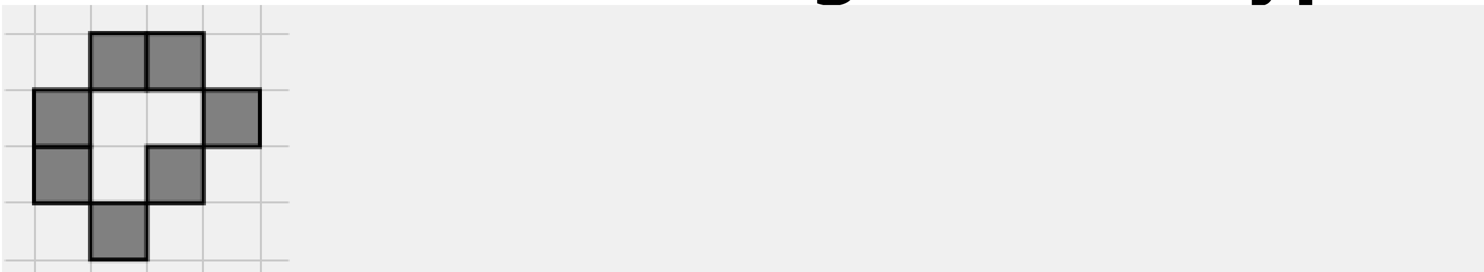
- The cellular automaton GOL = <2, {0,1}, Moore, f> is called a **game of life** (Conway, game of life, B3S23), where
  - 1. $f(1, b_2, \ldots, b_9) = 1$, ha $\sum_{2 \leq i \leq 9} b_i \in \{2,3\}$,
  - 2. $f(0, b_2, \ldots, b_9) = 1$, ha $\sum_{2 \leq i \leq 9} b_i = 3$,
  - 3. $f(b_1, b_2, \ldots, b_9) = 0$, otherwise.

- A living cell survives if it has 2 or 3 living Moore-neighbors,
  otherwise it dies due to isolation (0-1 neighbor) or overpopulation (4-8 neighbors).
- A dead cell becomes alive if and only if it has exactly 3 living Moore neighbors

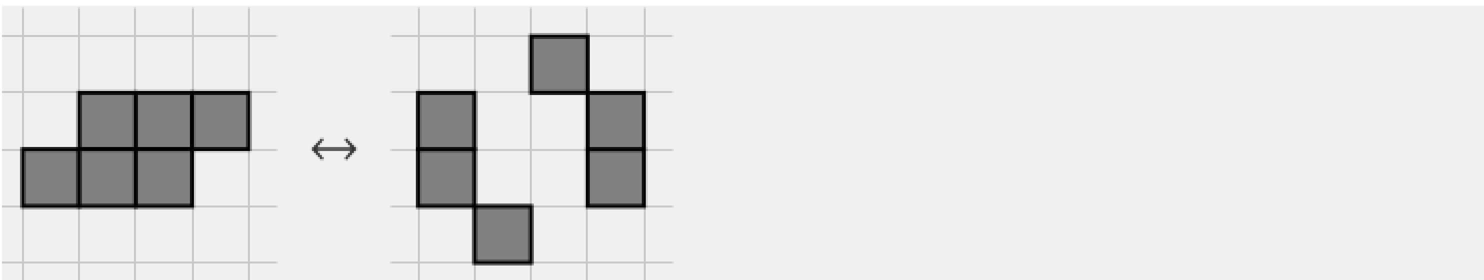
Game of Life – Configuration Types

- The **orbit** of a configuration c is the sequence
  orb(c) = c, $G(c)$, $G^2(c)$, . . .
  - c is a **still life** if $G(c) = c$.
    The smallest still life is a 2x2 block.
  - c is an **oscillator**, if $\exists\, i \geq 2$, then $G^i(c) = c$
    (c is periodic in time).
    Example: 3 living cells beside each other. This is the smallest (2-period) oscillator, called **blinker**.
  - c is a **spaceship**, if $\exists\, i \geq 1$, such that
    $\{x \in Z^2 \mid G^i(c)(x) = 1\} = \{x \in Z^2 \mid c(x) = 1\} + y$
    for some vector y.
    The smallest spaceship is the **glider**.
  - c is a **gun**, if $\exists\, i \geq 1$, such that $\forall\, k \in N$
    $\{x \in Z^2 \mid G^{(k+1)i}(c)(x) = 1\} \supset \{x \in Z^2 \mid G^{ki}(c)(x) = 1\}$ and
    $\{x \in Z^2 \mid G^{(k+1)i}(c)(x) = 1\} - \{x \in Z^2 \mid G^{ki}(c)(x) = 1\}$ is a spaceship.

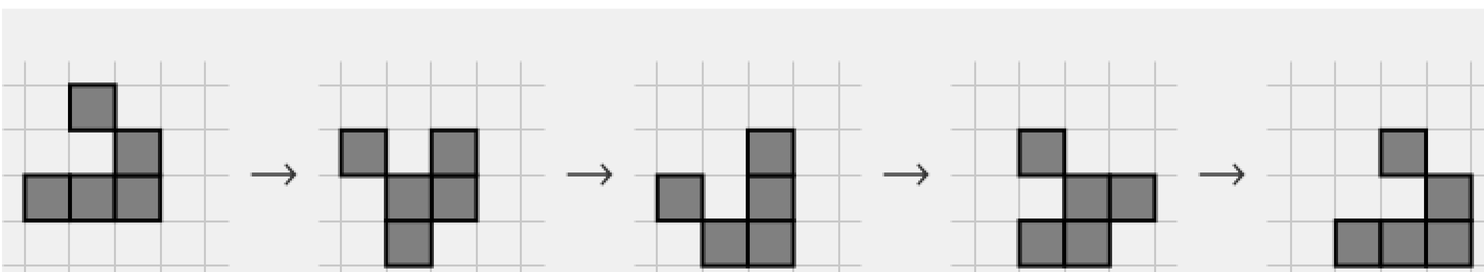
Game of Life – Configuration Types



Still life: Loaf



Oscillator: Toad

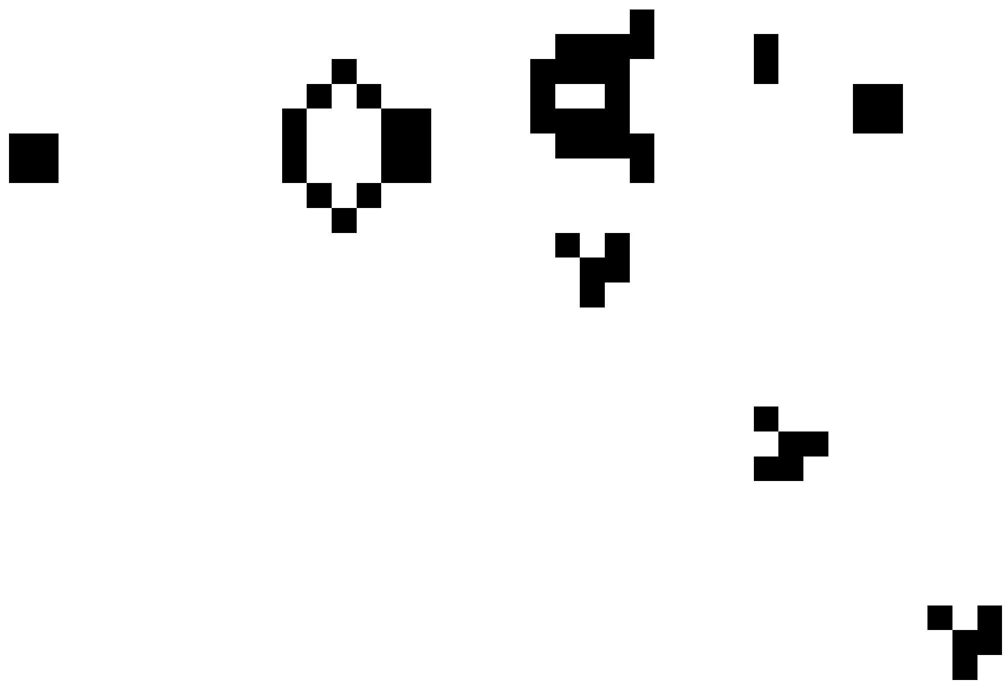


Spaceship: Glider

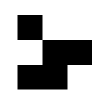
Game of Life – Configuration Types

Bill Gosper’s glider gun:

it produces the first glider in the 15th generation, then a new glider every 30th generation.



https://commons.wikimedia.org/w/index.php?curid=101736



One-dimensional, two-state cellular automaton

- Stephen Wolfram examined cellular automata of the form $A = \langle 1, \{0,1\}, (-1,0,+1), f \rangle$.
- There are 256 possibilities for the function $f : \{0,1\}^3 \rightarrow \{0,1\}$. These can be mapped to bit sequences of length 8, where the first bit is f(111), the second bit is f(110), . . . , the eighth bit is f(000). The decimal number corresponding to this mapping is called the Wolfram code of the cellular automaton. The corresponding cellular automaton is referred to by its Wolfram code (from 0 to 255).

- Example: What are the rules of the one-dimensional, two-state cellular automaton number 30? Write 30 as a binary number:

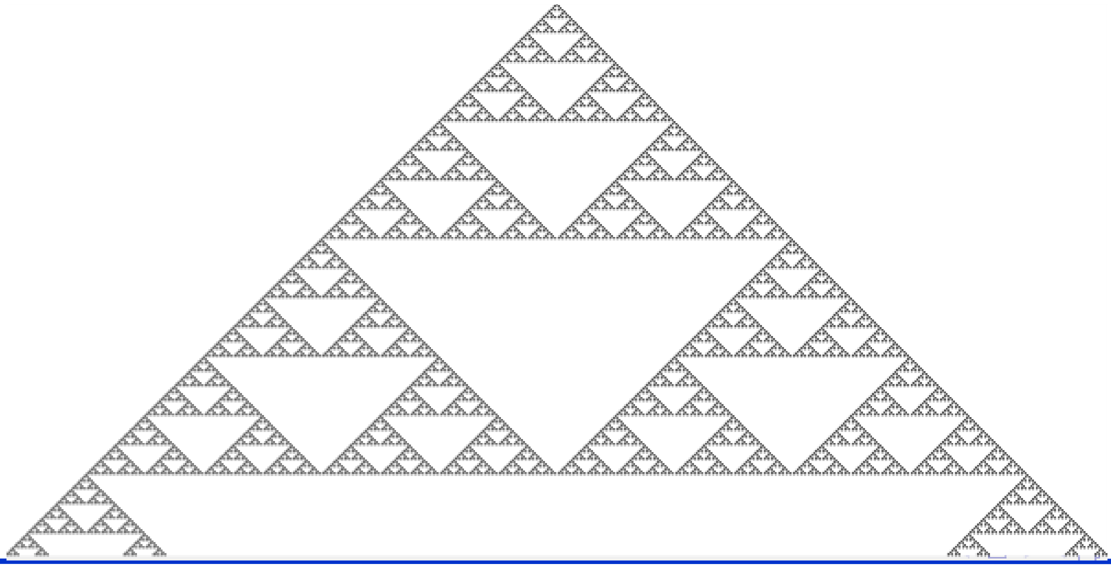
111	110	101	100	011	010	001	000
↓	↓	↓	↓	↓	↓	↓	↓
0	0	0	1	1	1	1	0

E.g., $f(0,1,1) = 1$ means that if the left neighbor of a living cell is dead and the right neighbor is alive, the cell remains alive.

One-dimensional, two-state cellular automaton

- The orbit of one-dimensional, two-state cellular automata can be represented with a 2-dimensional image, the so-called **space-time diagram**: the first row of the image represents the initial configuration c, and the i-th row represents $G^i(c)$.
- Example: rule 90

111	110	101	100	011	010	001	000
0	1	0	1	1	0	1	0



One-dimensional, two-state cellular automaton

- Wolfram classes:
- W1: The orbit of almost every initial configuration stabilizes in the same constant configuration. E.g. 160

111	110	101	100	011	010	001	000
1	0	1	0	0	0	0	0

- W2: The orbit of almost all initial configurations is periodic. E.g. 108

111	110	101	100	011	010	001	000
0	1	1	0	1	1	0	0

- W3: Almost every initial configuration leads to chaotic behavior. E.g. 126

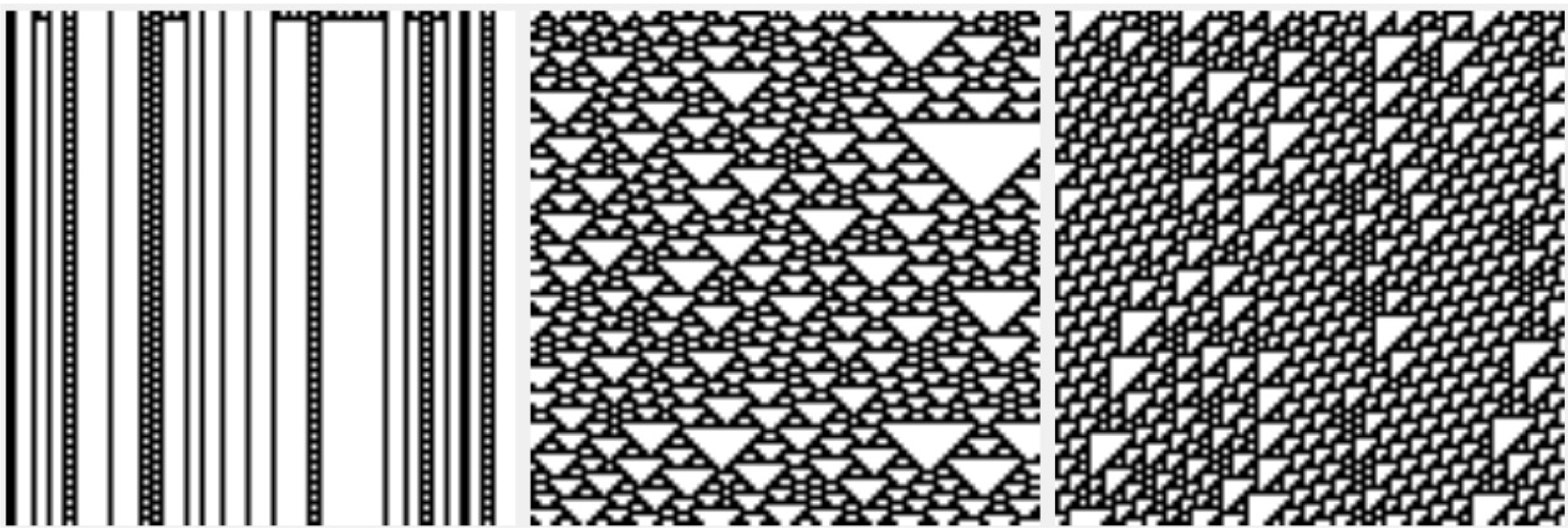
111	110	101	100	011	010	001	000
0	1	1	1	1	1	1	0

- W4: Local structures develop with complex relationships. E.g. 110

111	110	101	100	011	010	001	000
0	1	1	0	1	1	1	0

One-dimensional, two-state cellular automaton

- 108 (W2) 126 (W3) 110 (W4)

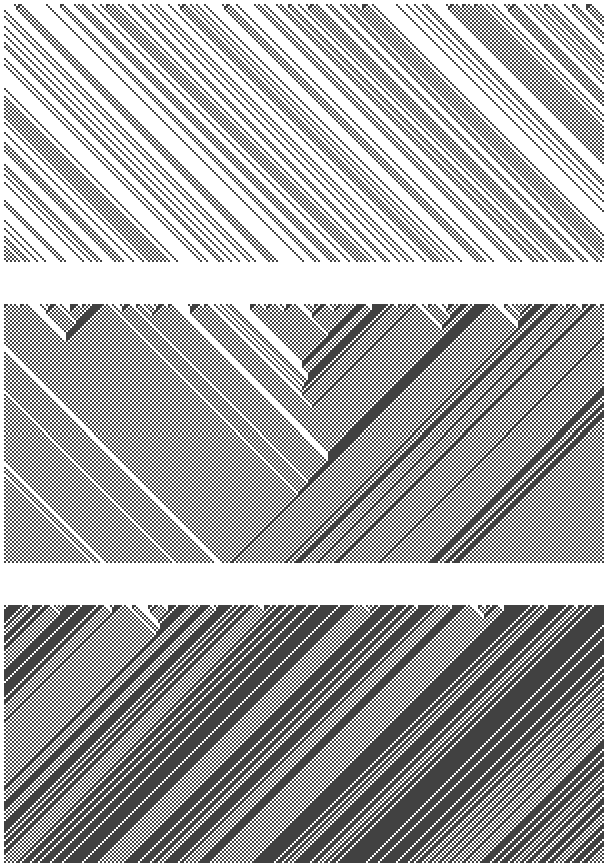


One-dimensional, two-state cellular automaton

- Rule 184

111	110	101	100	011	010	001	000
1	0	1	1	1	0	0	0

- Traffic flow. 1s are cars that move one step to the right if there is a free space in front of them (0).
  It models reality surprisingly well (continuous movement, traffic jams, stop-and-go).
  On the right side:
  25, 50, and 75 percent car density.

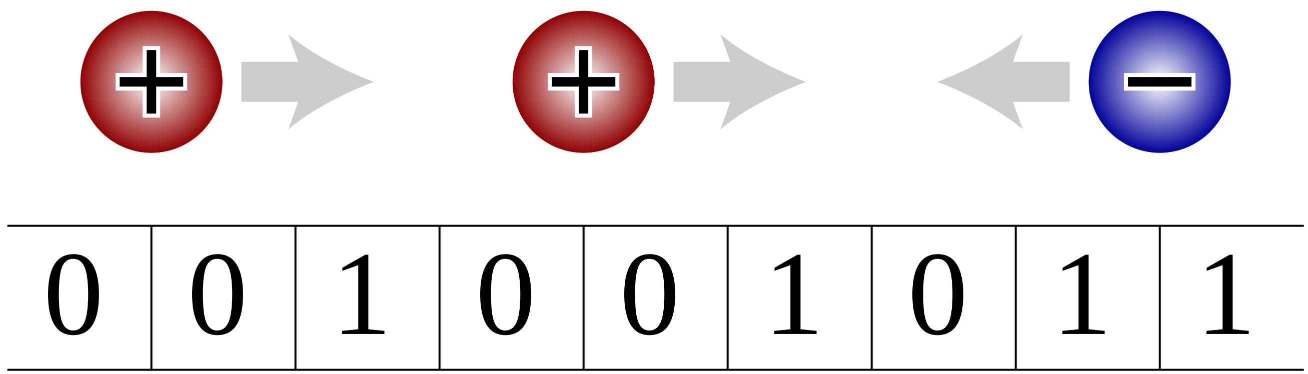


https://en.wikipedia.org/wiki/File:Rule_184.png

One-dimensional, two-state cellular automaton

- Rule 184

111	110	101	100	011	010	001	000
1	0	1	1	1	0	0	0



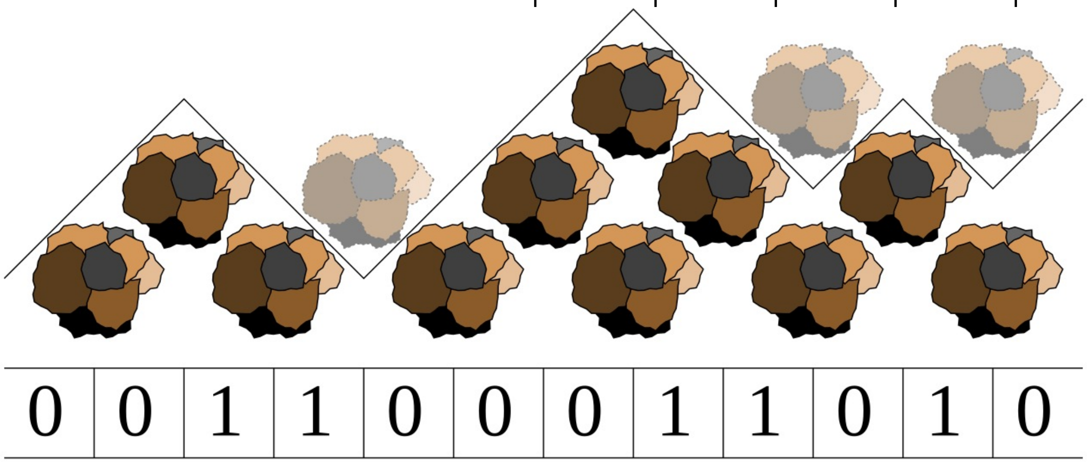
https://commons.wikimedia.org/w/index.php?curid=2455570

- Ballistic cancellation. The pattern 00 represents a positively charged particle moving from left to right, while the pattern 11 represents its antiparticle pair moving from right to left. 01 and 10 correspond to the intermediate space. The oppositely charged particle pairs cancel each other out.

One-dimensional, two-state cellular automaton

- Rule 184

111	110	101	100	011	010	001	000
1	0	1	1	1	0	0	0



https://commons.wikimedia.org/w/index.php?curid=2455569

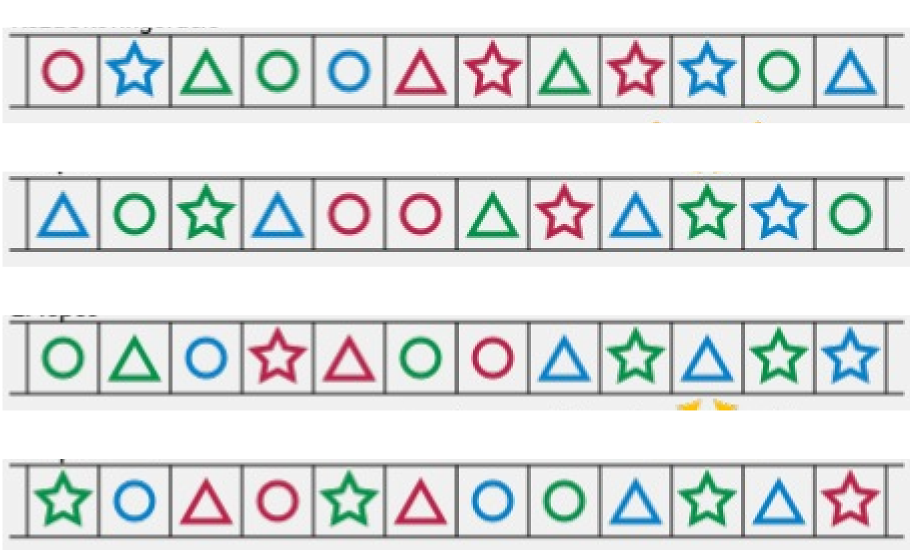
- Particle deposition on an irregular surface: Consider a grid rotated by 45-degree angle to gravity. We can model a surface so that each particle has a particle below it to the left and below it to the right. The surface of this is a boundary of line segments with slopes of +1 and -1. In every iteration, a new particle is deposited at the local minimum points (above 10).

Injective/Surjective/Bijective Cellular Automata

- A cellular automaton is **injective/surjective/bijective**, if G injective/surjective/bijective.
- That is, a cellular automaton is bijective if for every possible configuration c there is exactly one configuration c' for which G(c') = c.
- A cell state $p \in S$ is a **quiescent state** if $f(p, \dots, p) = p$. (A cell whose neighbors are all quiescent becomes quiescent.)
  If a configuration c has finitely many cells in a state different from the quiescent state p, then we say that c is a finite configuration of the cellular automaton.
- Remark: If c is finite, then G(c) is also finite.
- Let G_F denote the restriction of G to finite configurations.

Reversible Cellular Automata

- A cellular automaton with global transition function G is called **reversible** if it has an inverse, i.e. a cellular automaton with global transition function F such that $F \circ G = G \circ F = id$.
- Example: 1-dimensional cellular automaton with 9 states



- Remark: A cellular automaton is reversible if and only if it is bijective.

Garden of Eden

- A configuration of a cellular automaton is called a **Garden of Eden configuration** if it has no predecessor.
- A Garden of Eden can only occur as an initial configuration.
- Example: rule 110

111	110	101	100	011	010	001	000
0	1	1	0	1	1	1	0

Every configuration c containing 01010 is a garden of Eden. Indirectly, suppose that there is a c', such that G(c') = c. Consider the 0 in the middle. There are three cases depending on what the neighborhood of this cell in c' is.
1) 000, then the neighborhood of the previous one in c' is $*00$, which is impossible.
2) 111, then because of the previous 1, there is a 0 before it in c', but then the first 0 in the pattern is created from $*01$, which is impossible.
3) 100, then because of the next 1, there is a 1 after 100 in c', but then the last 0 in the pattern is created from $01*$, which is impossible.

Garden of Eden

- A finite pattern is called an **orphan** if every configuration containing it is a Garden of Eden.
- 01010 is an orphan in the one-dimensional, two-state cellular automaton with rule 110.
- **Theorem:** Any Garden of Eden contains an orphan.

Garden of Eden Theorem

- **Garden of Eden Theorem:** (Moore (1962), Myhill (1963))
  Let A = <X , S , N , f> be a cellular automaton, where X is a Euclidean space. Then G is surjective if and only if G_F is injective.

- That is, there is an orphan (Garden of Eden) if and only if there are two finite configurations that have the same image. Such pairs are called **twin**s.

- **Proof sketch:** We will prove it only in the Euclidean plane.
  - Let |S| = s. Suppose there are twins P and Q and suppose that they can be enclosed in an n × n square each.
  - Suppose further that the neighborhood of elements of X has radius at most n.
  - Consider a large square of size mn × mn.
    An area of at most (m+2)n × (m+2)n of the previous configuration can affect the state of these cells.
    This can be divided into (m+2) × (m+2) many n × n cells.

Garden of Eden Theorem

- **Proof (cont.):**
  - Since there are twins, at least 2 of the possible $s^{n \times n}$ configurations of these $n \times n$ parts have the same effect.
    Thus, at most $(s^{n \times n} - 1)^{(m+2)(m+2)}$ different images can be obtained out of the possible $s^{mn \times mn}$.
    The latter is the larger number, since $\log(s^{n^2})m^2 > \log(s^{n^2} - 1)(m^2 + 4m + 4)$ if m is large enough.

Garden of Eden Theorem

- **Proof (cont.):**
  - Suppose there is an orphan R. Then there exists $n \in N$ such that R can be enclosed in an $n \times n$ square.
  - Consider the finite configurations that can be contained in a large $mn \times mn$ square. The next generation of these configurations is contained in a $(m+2)n \times (m+2)n$ square, which can be divided into $(m+2) \times (m+2)$ many $n \times n$ squares.
  - Since we cannot get R in any of these, the potentially $s^{mn \times mn}$ many configuration can have at most $(s^{n \times n}-1)^{(m+2)(m+2)}$ many different images. Because of the previous calculation, according to the pigeonhole principle, there are two configurations that have the same image. □

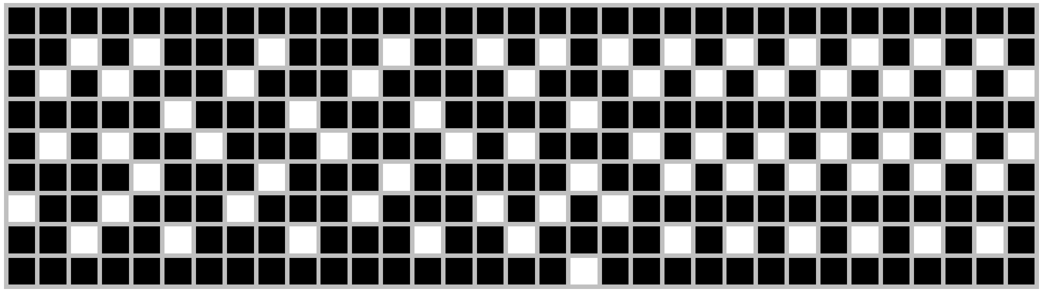
Garden of Eden Theorem

- **Corollaries:**
  - Every injective cellular automaton is also surjective.
  - It is equivalent to say that a cellular automaton is injective, bijective, reversible.
  - If G is injective, then G_F is surjective.

- G is injective ⇔ G is bijective ⇔ G is reversible ⇒
  ⇒ G_F is surjective ⇔ G_F is bijective ⇒
  ⇒ G is surjective ⇔ G_F is injective.

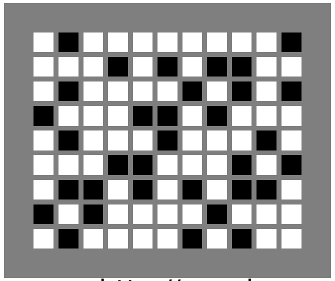
Garden of Eden Theorem

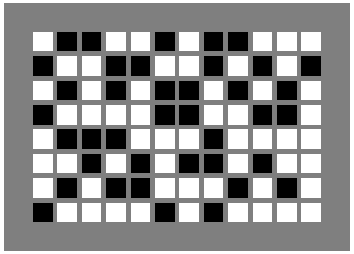
- The Garden of Eden Theorem can be applied to GOL. It is easy to see that there are twins. Let's take a configuration of all dead cells and a single living cell. So it follows that there is a Garden of Eden. However, GOL does not have a small Garden of Eden.
- Garden of Eden by Roger Banks (9x33):



https://commons.wikimedia.org/w/index.php?curid=2822838

- Steven Eker’s orphants (9x11 and 8x12):





http://wwwhomes.uni-bielefeld.de/achim/orphan_9th.html

Computation Power of Cellular Automata

- Let $A = \langle 1, Q, (-1,0,+1), f \rangle$ be a one-dimensional cellular automaton, $T \subset Q$, $F \subset Q \setminus T$, and $\_ \in Q \setminus (T \cap F)$ be the quiescent state of A.
  A **recognizes** the language $L \subseteq T^*$ if
  $w \in L \Leftrightarrow$ starting A with w (i.e., w can be read on consecutive cells, all other cells are quiescent) the cell corresponding to the first letter of w reaches a state in F.

- **Theorem:** Given a Turing machine M with n states and a tape alphabet of m elements. There is a one-dimensional cellular automaton that can simulate M with a six-element neighborhood and $\max\{n,m\} + 1$ states.

Game of Life – Undecidable Problems

- The Universal Turing Machine can be simulated in the Game of Life.

- **Theorem:** For every pair <M,w> (Turing machine, word), an initial configuration $c_{<M,w>}$ can be constructed in GOL, such that $c_{<M,w>}$ dies out $\Leftrightarrow$ $w \in L(M)$.

- **Theorem:** Given a configuration c and a configuration c' in GOL. It is undecidable whether there exists $i \geq 0$ such that $c' = G^i(c)$.

- **Theorem:** It is undecidable whether a finite configuration of GOL dies out.

Decidable and Undecidable Problems

- **Theorem:**
  1) It is decidable whether a 1-dimensional cellular automaton is injective.
  2) It is decidable whether a 1-dimensional cellular automaton is surjective.
- **Theorem:**
  1) It is undecidable whether a 2-dimensional cellular automaton is injective.
  2) It is undecidable whether a 2-dimensional cellular automaton is surjective.
  3) The injectivity of a 2-dimensional cellular automaton is recursively enumerable.
  4) The non-surjectivity of a 2-dimensional cellular automaton is recursively enumerable.

Undecidable Problems

- A **configuration is quiescent** if all cells are quiescent.
- A cellular automaton is **nilpotent** if it reaches a quiescent configuration for every initial configuration.

- **Theorem:**
  - It is undecidable whether a 1-dimensional cellular automaton is nilpotent.
  - It is undecidable whether a 2-dimensional cellular automaton is nilpotent.
  - It is recursively enumerable whether a 1-dimensional cellular automaton is nilpotent.
  - It is recursively enumerable whether a 2-dimensional cellular automaton is nilpotent.

Undecidable Problems

- **Theorem:**
  - It is undecidable whether a 2-dimensional cellular automaton is reversible.
  - It is recursively enumerable whether a 2-dimensional cellular automaton is reversible.

- A cellular automaton with global transition function G is **periodic** if for all initial configurations G is periodic.

- **Theorem:**
  - It is undecidable whether a 2-dimensional cellular automaton is periodic.
  - It is undecidable whether a 1-dimensional cellular automaton is periodic.