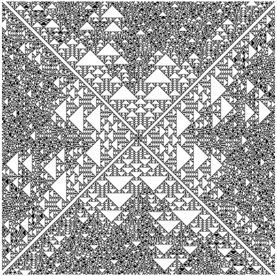
Cellular Automata, Lecture 2



Math Circles 2018
University of Waterloo

Part I

Multistate and Totalistic Automata

Multistate automata

Multistate automata

Last time: only allowed cells to be ALIVE or DEAD (1 or 0).

Multistate automata

Last time: only allowed cells to be ALIVE or DEAD (1 or 0).
These are called the **elementary cellular automata**, and there are 256 of them.

Multistate automata

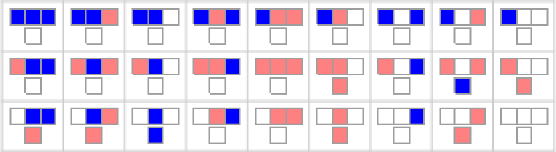
Last time: only allowed cells to be ALIVE or DEAD (1 or 0). These are called the **elementary cellular automata**, and there are 256 of them.

What about three states?

Multistate automata

Last time: only allowed cells to be ALIVE or DEAD (1 or 0). These are called the **elementary cellular automata**, and there are 256 of them.

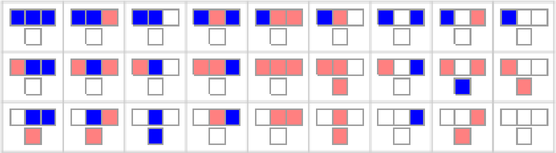
What about three states?

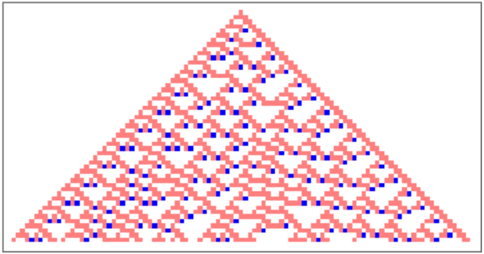


Multistate automata

Last time: only allowed cells to be ALIVE or DEAD (1 or 0). These are called the **elementary cellular automata**, and there are 256 of them.

What about three states?





Multistate automata

Some interesting 3-color automata:

- Rule 679458
- Rule 3333333
- Rule 21111

Multistate automata

Some interesting 3-color automata:

- Rule 679458
- Rule 3333333
- Rule 21111

```
CellularAutomaton[{679458,3}]
```

Multistate automata

Some interesting 3-color automata:

- Rule 679458
- Rule 3333333
- Rule 21111

```
CellularAutomaton[{679458,3}]
```

How many 3-color rules are there?

Multistate automata

Some interesting 3-color automata:

- Rule 679458
- Rule 3333333
- Rule 21111

```
CellularAutomaton[{679458,3}]
```

How many 3-color rules are there?

How many k-color rules are there?

Totalistic automata

Totalistic rules: the state of a cell only depends on the *total* of its neighboring cells.

Totalistic automata

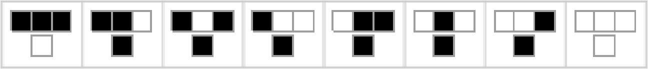
Totalistic rules: the state of a cell only depends on the *total* of its neighboring cells.

Total	3	2	1	0
State	0	1	1	0

Totalistic automata

Totalistic rules: the state of a cell only depends on the *total* of its neighboring cells.

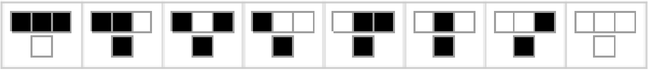
Total	3	2	1	0
State	0	1	1	0



Totalistic automata

Totalistic rules: the state of a cell only depends on the *total* of its neighboring cells.

Total	3	2	1	0
State	0	1	1	0

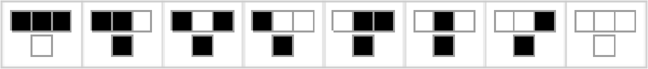


It's Rule 126! The friendship rule!

Totalistic automata

Totalistic rules: the state of a cell only depends on the *total* of its neighboring cells.

Total	3	2	1	0
State	0	1	1	0



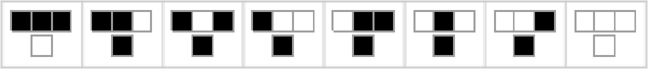
It's Rule 126! The friendship rule!

```
CellularAutomaton[{6, {2, 1}}]
```

Totalistic automata

Totalistic rules: the state of a cell only depends on the *total* of its neighboring cells.

Total	3	2	1	0
State	0	1	1	0



It's Rule 126! The friendship rule!

```
CellularAutomaton[{6, {2, 1}}]
```

How many 2-color totalistic rules are there?

Totalistic automata

What about for 3-color automata?

Totalistic automata

What about for 3-color automata?

CellularAutomaton[{2049, {3, 1}}]

Totalistic automata

What about for 3-color automata?

CellularAutomaton[{2049, {3, 1}}]

Total	6	5	4	3	2	1	0
State	2	2	1	0	2	2	0

Totalistic automata

What about for 3-color automata?

CellularAutomaton[{2049, {3, 1}}]

Total	6	5	4	3	2	1	0
State	2	2	1	0	2	2	0

Totalistic automata

What about for 3-color automata?

CellularAutomaton[{2049, {3, 1}}]

Total	6	5	4	3	2	1	0
State	2	2	1	0	2	2	0

Notice 2049 is just 2210220 in base-3.

Totalistic automata

What about for 3-color automata?

CellularAutomaton[{2049, {3, 1}}]

Total	6	5	4	3	2	1	0
State	2	2	1	0	2	2	0

Notice 2049 is just 2210220 in base-3.

How many 3-color totalistic rules are there?

Totalistic automata

What about for 3-color automata?

CellularAutomaton[{2049, {3, 1}}]

Total	6	5	4	3	2	1	0
State	2	2	1	0	2	2	0

Notice 2049 is just 2210220 in base-3.

How many 3-color totalistic rules are there?

How many *k*-color totalistic rules are there?

Totalistic automata: A Code Challenge

Exercise: Using Wolfram Language, write code that generates a 4-color totalistic rule with initial state 102321.

```
CellularAutomaton[{30,{4,1}},{{1,0,2,3,2,1},0},100]
```

Change 30 to another rule. See if you can find one that looks cool! Then use RulePlot to print out its transition rules.

Part II

The Mathematical Formalism of CA's

Mathematical definition?

In order to make *precise statements* about cellular automata, we need to form a *rigorous foundation*.

Mathematical definition?

In order to make *precise statements* about cellular automata, we need to form a *rigorous foundation*.

We should be able to provide a *rigorous explanation* for each of the following:

Mathematical definition?

In order to make *precise statements* about cellular automata, we need to form a *rigorous foundation*.

We should be able to provide a *rigorous explanation* for each of the following:

- What is a **state**?

Mathematical definition?

In order to make *precise statements* about cellular automata, we need to form a *rigorous foundation*.

We should be able to provide a *rigorous explanation* for each of the following:

- What is a **state**?
- What is a **neighborhood**?

Mathematical definition?

In order to make *precise statements* about cellular automata, we need to form a *rigorous foundation*.

We should be able to provide a *rigorous explanation* for each of the following:

- What is a **state**?
- What is a **neighborhood**?
- What do 1D and 2D automata have in common?

Suppose X is a set, $f : X \to X$ is a function, and $x_0 \in X$ is some point.

Then you can form a sequence

$$x_0,$$

Iterated function systems

Suppose X is a set, $f : X \to X$ is a function, and $x_0 \in X$ is some point.

Then you can form a sequence

$$x_0, \quad f(x_0),$$

Iterated function systems

Suppose X is a set, $f : X \to X$ is a function, and $x_0 \in X$ is some point.

Then you can form a sequence

$$x_0, \quad f(x_0), \quad f(f(x_0)),$$

Iterated function systems

Suppose X is a set, $f : X \to X$ is a function, and $x_0 \in X$ is some point.

Then you can form a sequence

$$x_0, \quad f(x_0), \quad f(f(x_0)), \quad f(f(f(x_0))),$$

Iterated function systems

Suppose X is a set, $f : X \to X$ is a function, and $x_0 \in X$ is some point.

Then you can form a sequence

$$x_0, \quad f(x_0), \quad f(f(x_0)), \quad f(f(f(x_0))), \ldots, f^n(x_0), \ldots$$

Iterated function systems

Suppose X is a set, $f : X \to X$ is a function, and $x_0 \in X$ is some point.

Then you can form a sequence

$$x_0, \quad f(x_0), \quad f(f(x_0)), \quad f(f(f(x_0))), \ldots, f^n(x_0), \ldots$$

In general, the nth term is $f^n(x_0)$ which is obtained by **iterating** f on the initial value x_0.

Iterated function systems

Suppose X is a set, $f : X \to X$ is a function, and $x_0 \in X$ is some point.

Then you can form a sequence

$$x_0, \quad f(x_0), \quad f(f(x_0)), \quad f(f(f(x_0))), \ldots, f^n(x_0), \ldots$$

In general, the nth term is $f^n(x_0)$ which is obtained by **iterating** f on the initial value x_0.

This sequence is called the **orbit** of x_0.

Example: The Collatz Sequence

Consider the function $f : \mathbf{N} \to \mathbf{N}$ given as follows.

$$f(n) := \begin{cases} 3n+1 & \text{if } n \text{ is odd,} \\ n/2 & \text{if } n \text{ is even.} \end{cases}$$

Starting at the initial value $n = 17$, you get the sequence

17,

Consider the function $f : \mathbf{N} \to \mathbf{N}$ given as follows.

$$f(n) := \begin{cases} 3n+1 & \text{if } n \text{ is odd,} \\ n/2 & \text{if } n \text{ is even.} \end{cases}$$

Starting at the initial value $n = 17$, you get the sequence

$$17, 52,$$

Example: The Collatz Sequence

Consider the function $f : \mathbf{N} \to \mathbf{N}$ given as follows.

$$f(n) := \begin{cases} 3n+1 & \text{if } n \text{ is odd,} \\ n/2 & \text{if } n \text{ is even.} \end{cases}$$

Starting at the initial value $n = 17$, you get the sequence

$$17, 52, 26,$$

Example: The Collatz Sequence

Consider the function $f : \mathbf{N} \to \mathbf{N}$ given as follows.

$$f(n) := \begin{cases} 3n + 1 & \text{if } n \text{ is odd,} \\ n/2 & \text{if } n \text{ is even.} \end{cases}$$

Starting at the initial value $n = 17$, you get the sequence

$$17, 52, 26, 13,$$

Example: The Collatz Sequence

Consider the function $f : \mathbf{N} \to \mathbf{N}$ given as follows.

$$f(n) := \begin{cases} 3n+1 & \text{if } n \text{ is odd,} \\ n/2 & \text{if } n \text{ is even.} \end{cases}$$

Starting at the initial value $n = 17$, you get the sequence

$$17, 52, 26, 13, 40,$$

Example: The Collatz Sequence

Consider the function $f : \mathbf{N} \to \mathbf{N}$ given as follows.

$$f(n) := \begin{cases} 3n+1 & \text{if } n \text{ is odd,} \\ n/2 & \text{if } n \text{ is even.} \end{cases}$$

Starting at the initial value $n = 17$, you get the sequence

$$17, 52, 26, 13, 40, 20,$$

Example: The Collatz Sequence

Consider the function $f : \mathbf{N} \to \mathbf{N}$ given as follows.

$$f(n) := \begin{cases} 3n+1 & \text{if } n \text{ is odd,} \\ n/2 & \text{if } n \text{ is even.} \end{cases}$$

Starting at the initial value $n = 17$, you get the sequence

$$17, 52, 26, 13, 40, 20, 10,$$

Example: The Collatz Sequence

Consider the function $f : \mathbf{N} \to \mathbf{N}$ given as follows.

$$f(n) := \begin{cases} 3n+1 & \text{if } n \text{ is odd,} \\ n/2 & \text{if } n \text{ is even.} \end{cases}$$

Starting at the initial value $n = 17$, you get the sequence

$$17, 52, 26, 13, 40, 20, 10, 5,$$

Example: The Collatz Sequence

Consider the function $f : \mathbf{N} \to \mathbf{N}$ given as follows.

$$f(n) := \begin{cases} 3n+1 & \text{if } n \text{ is odd,} \\ n/2 & \text{if } n \text{ is even.} \end{cases}$$

Starting at the initial value $n = 17$, you get the sequence

$$17, 52, 26, 13, 40, 20, 10, 5, 16,$$

Example: The Collatz Sequence

Consider the function $f : \mathbf{N} \to \mathbf{N}$ given as follows.

$$f(n) := \begin{cases} 3n+1 & \text{if } n \text{ is odd,} \\ n/2 & \text{if } n \text{ is even.} \end{cases}$$

Starting at the initial value $n = 17$, you get the sequence

$$17, 52, 26, 13, 40, 20, 10, 5, 16, 8,$$

Example: The Collatz Sequence

Consider the function $f : \mathbf{N} \to \mathbf{N}$ given as follows.

$$f(n) := \begin{cases} 3n+1 & \text{if } n \text{ is odd,} \\ n/2 & \text{if } n \text{ is even.} \end{cases}$$

Starting at the initial value $n = 17$, you get the sequence

$$17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4,$$

Example: The Collatz Sequence

Consider the function $f : \mathbf{N} \to \mathbf{N}$ given as follows.

$$f(n) := \begin{cases} 3n+1 & \text{if } n \text{ is odd,} \\ n/2 & \text{if } n \text{ is even.} \end{cases}$$

Starting at the initial value $n = 17$, you get the sequence

$$17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2,$$

Example: The Collatz Sequence

Consider the function $f : \mathbf{N} \to \mathbf{N}$ given as follows.

$$f(n) := \begin{cases} 3n+1 & \text{if } n \text{ is odd,} \\ n/2 & \text{if } n \text{ is even.} \end{cases}$$

Starting at the initial value $n = 17$, you get the sequence

$$17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1,$$

Example: The Collatz Sequence

Consider the function $f : \mathbf{N} \to \mathbf{N}$ given as follows.

$$f(n) := \begin{cases} 3n+1 & \text{if } n \text{ is odd,} \\ n/2 & \text{if } n \text{ is even.} \end{cases}$$

Starting at the initial value $n = 17$, you get the sequence

$$17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1, 4,$$

Example: The Collatz Sequence

Consider the function $f : \mathbf{N} \to \mathbf{N}$ given as follows.

$$f(n) := \begin{cases} 3n+1 & \text{if } n \text{ is odd,} \\ n/2 & \text{if } n \text{ is even.} \end{cases}$$

Starting at the initial value $n = 17$, you get the sequence

$$17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1, 4, 2,$$

Example: The Collatz Sequence

Consider the function $f : \mathbf{N} \to \mathbf{N}$ given as follows.

$$f(n) := \begin{cases} 3n+1 & \text{if } n \text{ is odd,} \\ n/2 & \text{if } n \text{ is even.} \end{cases}$$

Starting at the initial value $n = 17$, you get the sequence

$$17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1, 4, 2, 1, \ldots$$

Example: The Collatz Sequence

Consider the function $f : \mathbf{N} \to \mathbf{N}$ given as follows.

$$f(n) := \begin{cases} 3n+1 & \text{if } n \text{ is odd,} \\ n/2 & \text{if } n \text{ is even.} \end{cases}$$

Starting at the initial value $n = 17$, you get the sequence

$$17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1, 4, 2, 1, \ldots$$

This is called a **Collatz sequence**.

Example: The Collatz Sequence

Consider the function $f : \mathbf{N} \to \mathbf{N}$ given as follows.

$$f(n) := \begin{cases} 3n+1 & \text{if } n \text{ is odd,} \\ n/2 & \text{if } n \text{ is even.} \end{cases}$$

Starting at the initial value $n = 17$, you get the sequence

$$17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1, 4, 2, 1, \ldots$$

This is called a **Collatz sequence**.

Exercise: Find the Collatz sequence starting at 9.

Example: The Collatz Sequence

Millenium Problem: Prove that every Collatz sequence reaches the $4, 2, 1$ loop, no matter what the initial value.

You'll win $1,000,000!

Example: The Fibonacci Sequence

$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, \ldots$$

Example: The Fibonacci Sequence

$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, \ldots$$

It is defined as follows: $a_0 = a_1 = 1$, and

$$a_{n+2} = a_{n+1} + a_n.$$

Example: The Fibonacci Sequence

$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, \ldots$$

It is defined as follows: $a_0 = a_1 = 1$, and

$$a_{n+2} = a_{n+1} + a_n.$$

Here's another way to define it!

Example: The Fibonacci Sequence

$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, \ldots$$

It is defined as follows: $a_0 = a_1 = 1$, and

$$a_{n+2} = a_{n+1} + a_n.$$

Here's another way to define it!

$$f : \mathbf{R}^2 \to \mathbf{R}^2$$

$$f(a, b) = (a + b, a).$$

Example: The Fibonacci Sequence

$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, \ldots$$

It is defined as follows: $a_0 = a_1 = 1$, and

$$a_{n+2} = a_{n+1} + a_n.$$

Here's another way to define it!

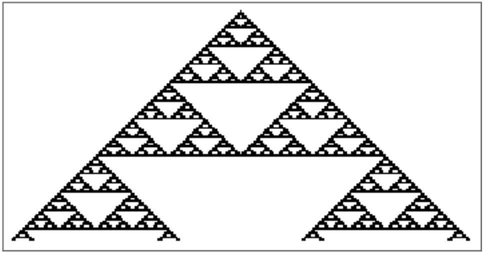
$$f : \mathbf{R}^2 \to \mathbf{R}^2$$

$$f(a, b) = (a + b, a).$$

Find the orbit of $(1, 1)$ under this function f.

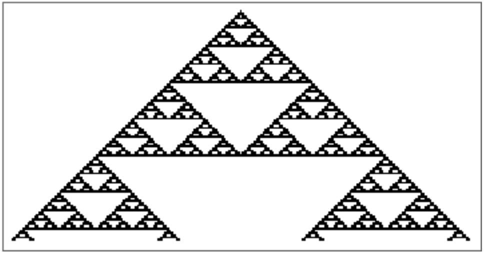
Cellular automata as iterated function systems

A cellular automaton starts with an initial value, and then we draw the *orbit* after iterating the rule.



Cellular automata as iterated function systems

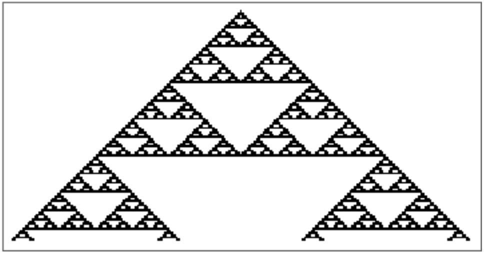
A cellular automaton starts with an initial value, and then we draw the *orbit* after iterating the rule.



- What is the "space" X?

Cellular automata as iterated function systems

A cellular automaton starts with an initial value, and then we draw the *orbit* after iterating the rule.



- What is the "space" X?
- What is the function $f : X \to X$?

What should a “state” be? (1D)

A state should assign 0 or 1 to each “cell”. The cells should be indexed by integers.

What should a "state" be? (1D)

A state should assign 0 or 1 to each "cell". The cells should be indexed by integers.

A **state** is just a function $X : \mathbf{Z} \to \{0, 1\}$.

What should a "state" be? (1D)

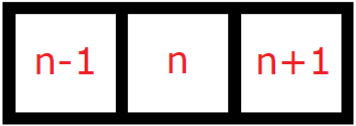
A state should assign 0 or 1 to each "cell". The cells should be indexed by integers.

A **state** is just a function $X : \mathbf{Z} \to \{0, 1\}$.

(For k colors, a state is a function $X : \mathbf{Z} \to \{1, 2, \ldots, k\}$.)

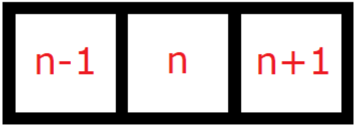
What should a “neighborhood” be? (1D)

The neighborhood of a cell consists of itself, and the two cells to its left and right.



What should a “neighborhood” be? (1D)

The neighborhood of a cell consists of itself, and the two cells to its left and right.



So the **neighborhood** of cell n is defined to be

$$\{n-1, n, n+1\}.$$

How should we define a 1D cellular automaton?

A cellular automaton should take a state x and produce a new state y.

How should we define a 1D cellular automaton?

A cellular automaton should take a state x and produce a new state y.

Common notation: for sets A, B, the set of functions from A to B is denoted

$$B^A := \{f : A \to B\}.$$

How should we define a 1D cellular automaton?

A cellular automaton should take a state x and produce a new state y.

Common notation: for sets A, B, the set of functions from A to B is denoted

$$B^A := \{f : A \to B\}.$$

So the set of states is $\{0,1\}^{\mathbf{Z}}$.

How should we define a 1D cellular automaton?

A cellular automaton should take a state x and produce a new state y.

Common notation: for sets A, B, the set of functions from A to B is denoted

$$B^A := \{f : A \to B\}.$$

So the set of states is $\{0,1\}^{\mathbf{Z}}$.

Thus, a **cellular automaton** is a function

$$\tau : \{0,1\}^{\mathbf{Z}} \to \{0,1\}^{\mathbf{Z}}$$

$$x \mapsto \tau(x).$$

How should we define a 1D cellular automaton?

A cellular automaton should take a state x and produce a new state y.

Common notation: for sets A, B, the set of functions from A to B is denoted

$$B^A := \{f : A \to B\}.$$

So the set of states is $\{0,1\}^{\mathbf{Z}}$.

Thus, a **cellular automaton** is a function

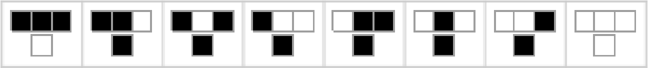
$$\tau : \{0,1\}^{\mathbf{Z}} \to \{0,1\}^{\mathbf{Z}}$$

$$x \mapsto \tau(x).$$

This is a function whose inputs and outputs are ALSO functions!

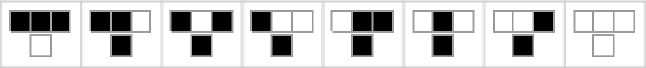
Example: Rule 126

Here are the transition rules for Rule 126.



Example: Rule 126

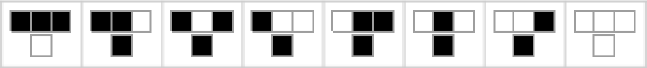
Here are the transition rules for Rule 126.



Let's write this as a function $\tau : \{0,1\}^{\mathbf{Z}} \to \{0,1\}^{\mathbf{Z}}$.

Example: Rule 126

Here are the transition rules for Rule 126.



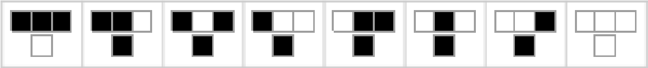
Let's write this as a function $\tau : \{0,1\}^{\mathbf{Z}} \to \{0,1\}^{\mathbf{Z}}$.

- Since Rule 126 is totalistic, we can define its **memory rule** by

$$\mu(a,b,c) := \begin{cases} 1 & \text{if } a+b+c \in \{0,3\} \\ & \end{cases}$$

Example: Rule 126

Here are the transition rules for Rule 126.



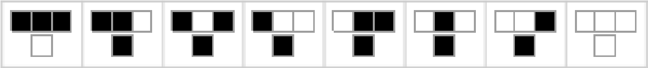
Let's write this as a function $\tau : \{0,1\}^{\mathbf{Z}} \to \{0,1\}^{\mathbf{Z}}$.

- Since Rule 126 is totalistic, we can define its **memory rule** by

$$\mu(a,b,c) := \begin{cases} 1 & \text{if } a+b+c \in \{0,3\} \\ 0 & \text{if } a+b+c \in \{1,2\} \end{cases}$$

Example: Rule 126

Here are the transition rules for Rule 126.



Let's write this as a function $\tau : \{0,1\}^{\mathbf{Z}} \to \{0,1\}^{\mathbf{Z}}$.

- Since Rule 126 is totalistic, we can define its **memory rule** by

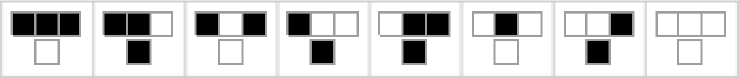
$$\mu(a,b,c) := \begin{cases} 1 & \text{if } a+b+c \in \{0,3\} \\ 0 & \text{if } a+b+c \in \{1,2\} \end{cases}$$

- For a state $x : \mathbf{Z} \to \{0,1\}$, define τ_x to be the state

$$\tau(x)_n := \mu(x_{n-1}, x_n, x_{n+1}).$$

Exercise For You: Rule 90

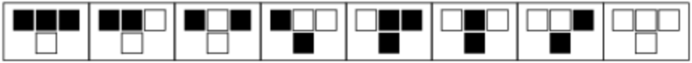
Here are the transition rules for Rule 90 (the XOR automaton).



Find the memory function μ for Rule 90, and write the cellular automaton τ in terms of μ.

Exercise For You, 2: Rule 30

Here are the transition rules for Rule 30.



Find the memory function μ for Rule 30, and write the cellular automaton τ in terms of μ.

(Hint: the memory function involves XOR.)

I lied about the definition of "cellular automaton"

Now we've seen that a cellular automaton τ should have a memory function.

I lied about the definition of "cellular automaton"

Now we've seen that a cellular automaton τ should have a memory function.

So the REAL definition of **cellular automaton** is a function

$$\tau : \{0,1\}^{\mathbf{Z}} \to \{0,1\}^{\mathbf{Z}}$$

which is determined by a **memory rule** $\mu : \{0,1\}^3 \to \{0,1\}$:

$$\tau(x)_n = \mu(x_{n-1}, x_n, x_{n+1}).$$

I lied about the definition of "cellular automaton"

Now we've seen that a cellular automaton τ should have a memory function.

So the REAL definition of **cellular automaton** is a function

$$\tau : \{0,1\}^{\mathbf{Z}} \to \{0,1\}^{\mathbf{Z}}$$

which is determined by a **memory rule** $\mu : \{0,1\}^3 \to \{0,1\}$:

$$\tau(x)_n = \mu(x_{n-1}, x_n, x_{n+1}).$$

(If you want k colors, replace $\{0,1\}$ by $\{1,\dots,k\}$.)

Part III

(A little bit on) 2-Dimensional Automata

Return to Planet Friendship

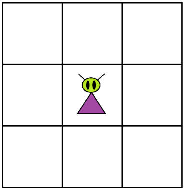
Return to Planet Friendship

Zoink is back!

Return to Planet Friendship

Zoink is back!

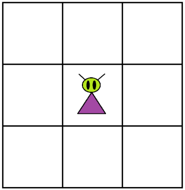
Zoink has a new power: he can grow friends **in all eight directions around them!**



Return to Planet Friendship

Zoink is back!

Zoink has a new power: he can grow friends **in all eight directions around them!**



This is called **2-dimensional friendification**.

Rules for 2-Dimensional Friendification

Rules for 2-Dimensional Friendification

1. **Loneliness:** If an alien has ≤ 1 live neighbors, he dies :(

Rules for 2-Dimensional Friendification

1. **Loneliness:** If an alien has ≤ 1 live neighbors, he dies :(
2. **Survival:** If an alien has two or three live neighbors, he survives :D

Rules for 2-Dimensional Friendification

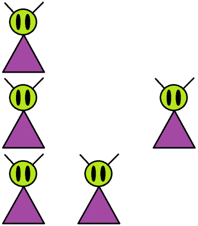
1. **Loneliness:** If an alien has ≤ 1 live neighbors, he dies :(
2. **Survival:** If an alien has two or three live neighbors, he survives :D
3. **Overpopulation:** If an alien has ≥ 4 live neighbors, he dies :(

Rules for 2-Dimensional Friendification

1. **Loneliness:** If an alien has ≤ 1 live neighbors, he dies :(
2. **Survival:** If an alien has two or three live neighbors, he survives :D
3. **Overpopulation:** If an alien has ≥ 4 live neighbors, he dies :(
4. **Birth:** If a cell has *exactly three* live neighbors, an alien will be born there!

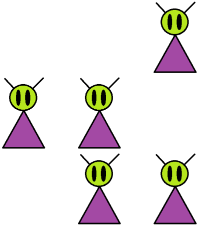
Exercise

Suppose the aliens stand in the following arrangement. Figure out who will live, die, or survive over the next two days.



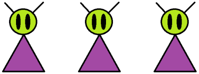
Exercise

Suppose the aliens stand in the following arrangement. Figure out who will live, die, or survive over the next two days.



More exercises

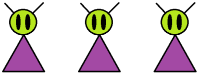
(a) For each of these two states, see what happens over the course of seven days.

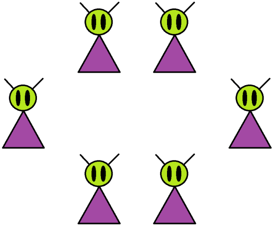


(b) Find an example of a state that never changes.

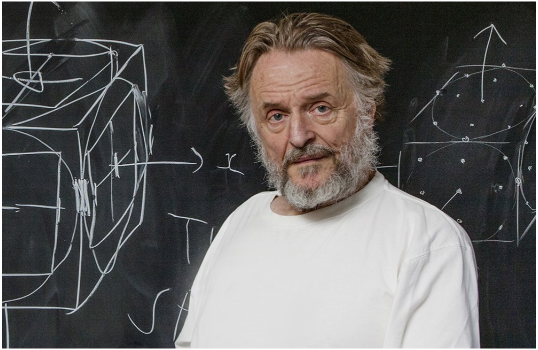
More exercises

(a) For each of these two states, see what happens over the course of seven days.





(b) Find an example of a state that never changes.



John Conway

[VIDEO]

A Useful App

http://conwaylife.appspot.com/new

2D Cellular Automata

Each cell has **eight** neighbors, which can be either alive or dead (black or white).

2D Cellular Automata

Each cell has **eight** neighbors, which can be either alive or dead (black or white).

- How many 2D cellular automata are there?
- If there are k colors, how many 2D cellular automata are there?
- How many 2D totalistic (two-state) automata are there?

Have a nice holiday!