Cellular Automata, Lecture 2

Math Circles 2018
University of Waterloo
Part I

Multistate and Totalistic Automata
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).
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?
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?
Some interesting 3-color automata:

- Rule 679458
- Rule 3333333
- Rule 21111
Some interesting 3-color automata:

- Rule 679458
- Rule 3333333
- Rule 21111

CellularAutomaton[{679458,3}]
Some interesting 3-color automata:

- Rule 679458
- Rule 3333333
- Rule 21111

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

How many 3-color rules are there?
Some interesting 3-color automata:

- Rule 679458
- Rule 333333
- 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 rules:** the state of a cell only depends on the total of its neighboring cells.

<table>
<thead>
<tr>
<th>Total</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Totalistic automata

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

<table>
<thead>
<tr>
<th>Total</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

It's Rule 126! The friendship rule!

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

How many 2-color totalistic rules are there?
Totalistic automata

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

<table>
<thead>
<tr>
<th>Total</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

It’s Rule 126! The friendship rule!
**Totalistic rules:** the state of a cell only depends on the total of its neighboring cells.

<table>
<thead>
<tr>
<th>Total</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

It’s Rule 126! The friendship rule!

```
CellularAutomaton[{6, {2, 1}}]
```
**Totalistic rules**: the state of a cell only depends on the total of its neighboring cells.

<table>
<thead>
<tr>
<th>Total</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

It’s Rule 126! The friendship rule!

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

How many 2-color totalistic rules are there?
What about for 3-color automata?
What about for 3-color automata?

```
CellularAutomaton[{2049, {3, 1}}]
```
What about for 3-color automata?

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

<table>
<thead>
<tr>
<th>Total</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>
Totalistic automata

What about for 3-color automata?

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

<table>
<thead>
<tr>
<th>Total</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

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

What about for 3-color automata?

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

<table>
<thead>
<tr>
<th>Total</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

Notice 2049 is just 2210220 in base-3.
What about for 3-color automata?

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

<table>
<thead>
<tr>
<th>Total</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

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}}]

<table>
<thead>
<tr>
<th>Total</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

Notice 2049 is just 2210220 in base-3.

How many 3-color totalistic rules are there?

How many $k$-color totalistic rules are there?
**Exercise:** Using Wolfram Language, write code that generates a 4-color totalistic rule with initial state 102321.

```wolfram
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*. 

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

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

Then you can form a sequence

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

$$x_0, \quad f(x_0), \quad f(f(x_0)),$$
Suppose $X$ is a set, $f : X \rightarrow X$ is a function, and $x_0 \in X$ is some point.

Then you can form a sequence

$$x_0, \quad f(x_0), \quad f(f(x_0)), \quad f(f(f(x_0))), \ldots$$
Suppose $X$ is a set, $f : X \rightarrow X$ is a function, and $x_0 \in X$ is some point.

Then you can form a sequence

$$x_0, \quad f(x_0), \quad f(f(x_0)), \quad f(f(f(x_0))), \ldots, f^n(x_0), \ldots$$
Suppose $X$ is a set, $f : X \rightarrow X$ is a function, and $x_0 \in X$ is some point.

Then you can form a sequence

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

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

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

In general, the $n$th 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 $x_0$. 

Iterated function systems
Consider the function $f : \mathbb{N} \to \mathbb{N}$ given as follows.

\[
f(n) :\begin{cases} 
3n + 1 & \text{if } n \text{ is odd}, \\
\frac{n}{2} & \text{if } n \text{ is even}.
\end{cases}
\]

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

17,
Example: The Collatz Sequence

Consider the function $f : \mathbb{N} \rightarrow \mathbb{N}$ 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,
Consider the function \( f : \mathbb{N} \rightarrow \mathbb{N} \) 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

\[17, 52, 26,\ldots\]

This is called a Collatz sequence.
Consider the function $f : \mathbb{N} \to \mathbb{N}$ 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

$$17, 52, 26, 13,$$
Consider the function $f : \mathbb{N} \to \mathbb{N}$ given as follows.

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

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

$$17, 52, 26, 13, 40,$$
Consider the function $f : \mathbb{N} \rightarrow \mathbb{N}$ 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

$$17, 52, 26, 13, 40, 20,$$
Consider the function $f : \mathbb{N} \rightarrow \mathbb{N}$ 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,
Consider the function $f : \mathbb{N} \rightarrow \mathbb{N}$ 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

$$17, 52, 26, 13, 40, 20, 10, 5,$$
Consider the function $f : \mathbb{N} \rightarrow \mathbb{N}$ 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 $f : \mathbb{N} \rightarrow \mathbb{N}$ 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 $f : \mathbb{N} \to \mathbb{N}$ 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

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

Consider the function $f : \mathbb{N} \rightarrow \mathbb{N}$ 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

$$17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2,$$
Consider the function $f : \mathbb{N} \rightarrow \mathbb{N}$ 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

$17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1,$
Consider the function $f : \mathbb{N} \to \mathbb{N}$ 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

$$17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1, 4,$$
Consider the function $f : \mathbb{N} \to \mathbb{N}$ 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

$$17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1, 4, 2,$$
Consider the function $f : \mathbb{N} \to \mathbb{N}$ 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 $f : \mathbb{N} \rightarrow \mathbb{N}$ 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.
Consider the function $f : \mathbb{N} \rightarrow \mathbb{N}$ 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.
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, ...
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, ...

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, ... 

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!

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

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

\[
f : \mathbb{R}^2 \rightarrow \mathbb{R}^2
\]

\[
f(a, b) = (a + b, a).
\]

Find the orbit of \((1, 1)\) under this function \(f\).
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” $X$?
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 \( f : X \rightarrow X \)?
A state should assign 0 or 1 to each "cell". The cells should be indexed by integers.
A state should assign 0 or 1 to each “cell”. The cells should be indexed by integers.

A state is just a function $X : \mathbb{Z} \rightarrow \{0, 1\}$. 

\[ ... -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 \frac{1}{2} ... \]
A state should assign 0 or 1 to each “cell”. The cells should be indexed by integers.

\[ \ldots \, -b \, -5 \, -4 \, -3 \, -2 \, -1 \, 0 \, 1 \, 2 \, 3 \, 4 \, 5 \, b \ldots \]

A state is just a function \( X : \mathbb{Z} \to \{0, 1\} \).

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

$$\{n - 1, n, n + 1\}.$$
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 $\{0, 1\}^\mathbb{Z}$. 

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 $\{0, 1\}^\mathbb{Z}$.

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

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

$$x \mapsto \tau(x).$$
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 $\{0, 1\}^\mathbb{Z}$.

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

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

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

This is a function whose inputs and outputs are ALSO functions!
Example: Rule 126

Here are the transition rules for Rule 126.

\[
\begin{array}{cccccc}
\text{\bf 126} & \text{\bf 126} & \text{\bf 126} & \text{\bf 126} & \text{\bf 126} & \text{\bf 126} \\
\hline \\
\text{\bf 126} & \text{\bf 126} & \text{\bf 126} & \text{\bf 126} & \text{\bf 126} & \text{\bf 126} \\
\end{array}
\]
Here are the transition rules for Rule 126.

Let’s write this as a function \( \tau : \{0, 1\}^\mathbb{Z} \rightarrow \{0, 1\}^\mathbb{Z} \).
Example: Rule 126

Here are the transition rules for Rule 126.

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

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

$\mu(a, b, c) := \begin{cases} 1 & \text{if } a + b + c \in \{0, 3\} \\ \end{cases}$
Here are the transition rules for Rule 126.

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

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

$$
\mu(a, b, c) := \begin{cases} 
1 & \text{if } a + b + c \in \{0, 3\} \\
0 & \text{if } a + b + c \in \{1, 2\}
\end{cases}
$$
Here are the transition rules for Rule 126.

Let’s write this as a function \( \tau : \{0, 1\}^Z \rightarrow \{0, 1\}^Z \).

- 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 \( x : Z \rightarrow \{0, 1\} \), define \( \tau_x \) 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 $\mu$ for Rule 90, and write the cellular automaton $\tau$ in terms of $\mu$. 
Here are the transition rules for Rule 30.

Find the memory function $\mu$ for Rule 30, and write the cellular automaton $\tau$ in terms of $\mu$.

(Hint: the memory function involves XOR.)
Now we’ve seen that a cellular automaton $\tau$ should have a memory function.
I lied about the definition of “cellular automaton”

Now we’ve seen that a cellular automaton $\tau$ should have a memory function.

So the REAL definition of cellular automaton is a function

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

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

$$\tau(x)_n = \mu(x_{n-1}, x_n, x_{n+1}).$$
Now we’ve seen that a cellular automaton $\tau$ should have a memory function.

So the REAL definition of cellular automaton is a function

$$\tau : \{0, 1\}^Z \rightarrow \{0, 1\}^Z$$

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

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

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

(A little bit on) 2-Dimensional Automata
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.
Zoink is back!
Zoink is back!

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

1. Loneliness: If an alien has \[ \leq 1 \] live neighbors, he dies :(

2. Survival: If an alien has two or three live neighbors, he survives :D

3. Overpopulation: If an alien has \[ \geq 4 \] live neighbors, he dies :(

4. Birth: If a cell has exactly three live neighbors, an alien will be born there!
1. **Loneliness:** If an alien has \( \leq 1 \) live neighbors, he dies :(

2. **Survival:** If an alien has two or three live neighbors, he survives :D

3. **Overpopulation:** If an alien has \( \geq 4 \) live neighbors, he dies :(

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

2. **Survival:** If an alien has two or three live neighbors, he survives :D

3. **Overpopulation:** If an alien has \( \geq 4 \) live neighbors, he dies :(
1. **Loneliness:** If an alien has $\leq 1$ live neighbors, he dies :(

2. **Survival:** If an alien has two or three live neighbors, he survives :D

3. **Overpopulation:** If an alien has $\geq 4$ 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.
(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.
(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.
Conway’s Game Of Life

John Conway

[VIDEO]
A Useful App

http://conwaylife.appspot.com/new
Each cell has eight neighbors, which can be either alive or dead (black or white).
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!