

Assignment

Cellular Automata (CA)

<https://www.artificial-intelligence.video/cellular-automata-and-digital-art>

The process of a CA that exhibits behavior similar to biological reproduction and evolution.

The cells live on a ***grid***.

Each cell has a ***state***.

Each cell has a ***neighborhood***.

a grid of cells, each "on" or "off"

a neighborhood
of cells

off	off	on	off	on	on
on	off	off	off	on	on
on	off	on	on	on	off
off	off	on	off	on	on
on	on	off	off	on	off
on	on	on	off	off	on
on	off	off	on	on	on
off	off	on	off	on	off

Wolfram's elementary CA

1) **Grid.** The simplest grid would be one-dimensional: a line of cells.



Figure 7.2

2) **States.** The simplest set of states (beyond having only one state) would be two states: 0 or 1.

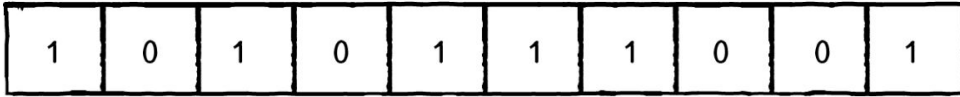


Figure 7.3

3) **Neighborhood.** The simplest neighborhood in one dimension for any given cell would be the cell itself and its two adjacent neighbors: one to the left and one to the right.

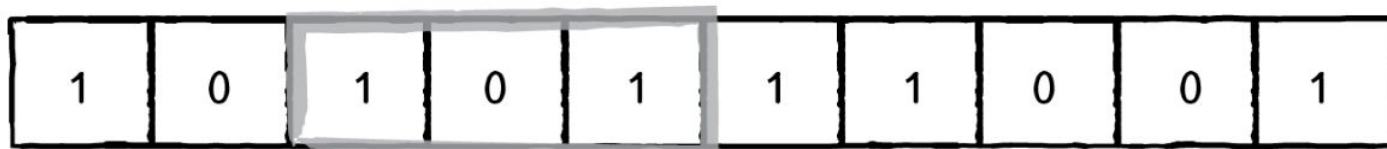


Figure 7.4: A neighborhood is three cells.

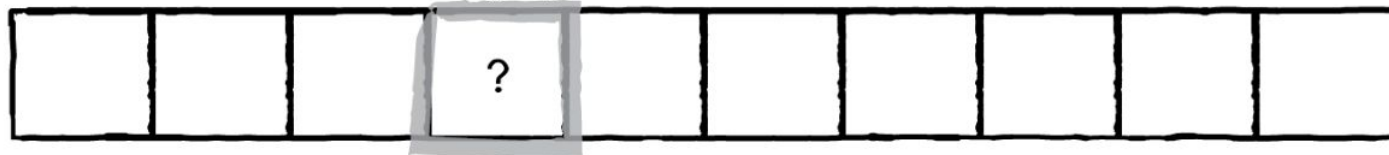
1	0	1	0	1	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---

[illegible][illegible]

generation 0



generation 1



Possible Combinations	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
	↓	↓	↓	↓	↓	↓	↓	↓
Rule Set	0	1	0	1	1	0	1	0

In terms of a Wolfram elementary CA,
there are 256 possible rulesets.

For every cell in the array:

- Take a look at the neighborhood states: left, middle, right.
- Look up the new value for the cell state according to some ruleset.
- Set the cell's state to that new value.

```
int[] cells = {1,0,1,0,0,0,0,1,0,1,1,1,0,0,0,1,1,1,0,0}
```

```
// For every cell in the array...
```

```
for (int i = 0; i < cells.length; i++) {
```

```
    // ...take a look at the neighborhood.
```

```
    int left    = cell[i-1];
```

```
    int middle = cell[i];
```

```
    int right   = cell[i+1];
```

```
    //[end]
```

```
// Look up the new value according to the rules.
```

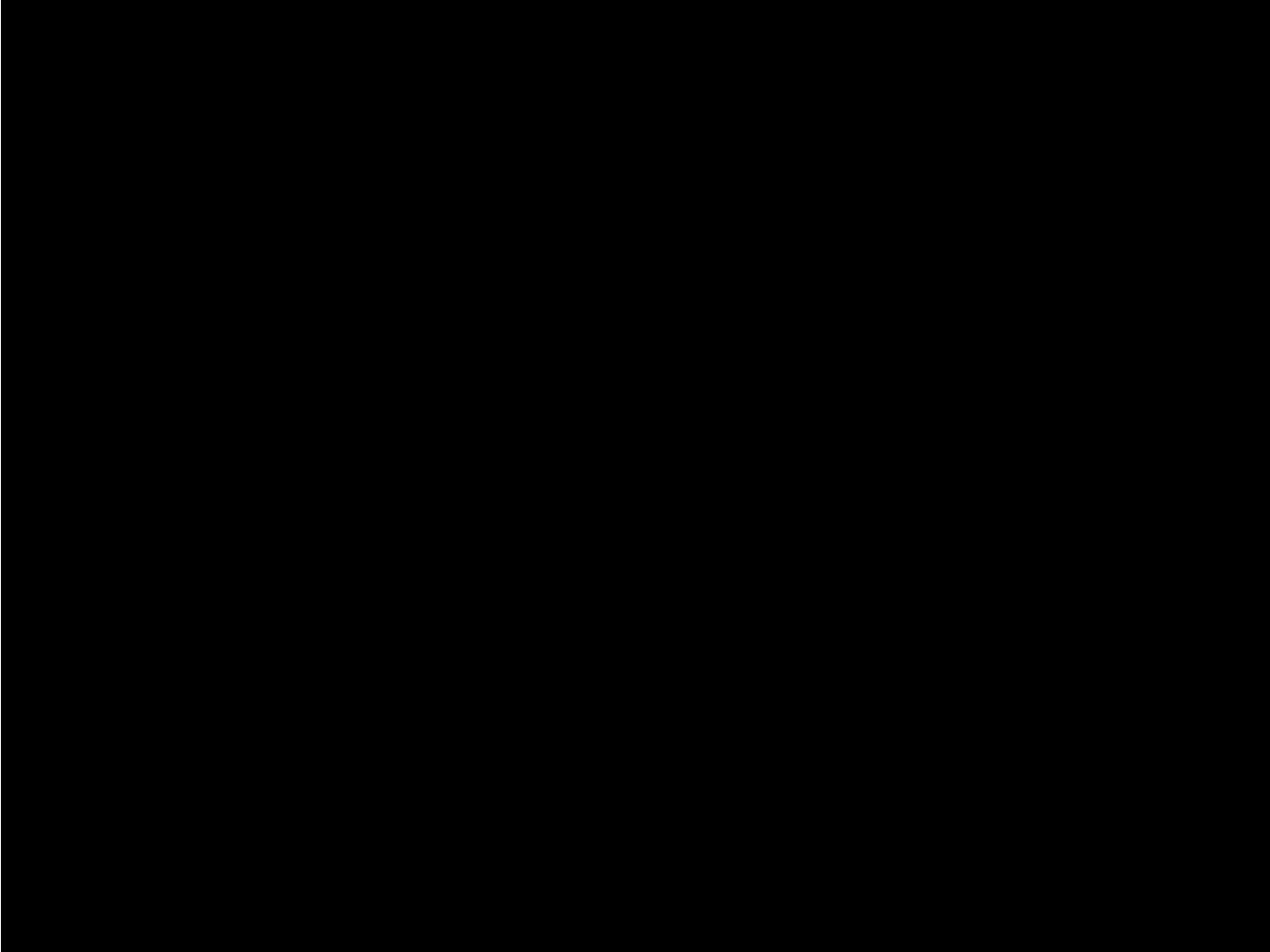
```
int newstate = rules(left,middle,right);
```

```
// Set the cell's state to the new value.
```

```
cell[i] = newstate;
```

```
}
```

```
int rules (int a, int b, int c) {  
    if      (a == 1 && b == 1 && c == 1) return ruleset[0];  
    else if (a == 1 && b == 1 && c == 0) return ruleset[1];  
    else if (a == 1 && b == 0 && c == 1) return ruleset[2];  
    else if (a == 1 && b == 0 && c == 0) return ruleset[3];  
    else if (a == 0 && b == 1 && c == 1) return ruleset[4];  
    else if (a == 0 && b == 1 && c == 0) return ruleset[5];  
    else if (a == 0 && b == 0 && c == 1) return ruleset[6];  
    else if (a == 0 && b == 0 && c == 0) return ruleset[7];  
    return 0;  
}
```

Two dimensional CA

(The Game of Life)

Two-dimensional cellular automata

1	0	1	0	1	0
0	0	1	0	1	1
1	1	1	0	1	1
1	0	1	0	1	0
0	0	0	1	1	0
1	1	0	0	1	0
1	1	1	0	0	0
1	0	1	1	1	1

a neighborhood
of 9 cells

1. **Death.** If a cell is alive (state = 1) it will die (state becomes 0) under the following circumstances.
 - **Overpopulation:** If the cell has four or more alive neighbors, it dies.
 - **Loneliness:** If the cell has one or fewer alive neighbors, it dies.
2. **Birth.** If a cell is dead (state = 0) it will come to life (state becomes 1) if it has exactly three alive neighbors (no more, no less).
3. **Stasis.** In all other cases, the cell state does not change. To be thorough, let's describe those scenarios.
 - **Staying Alive:** If a cell is alive and has exactly two or three live neighbors, it stays alive.
 - **Staying Dead:** If a cell is dead and has anything other than three live neighbors, it stays dead.

Demo

Want to explore more?

<https://natureofcode.com/book/chapter-7-cellular-automata/>

Want to Learn Advance Tricks?

Creative Programming 2

Image processing, interactivity, and computer vision etc.

Interactive Installations

Arduino and digital fabrication
(Instructor: Jeffrey Thompson)

PYTHON

Terminal

the application provides text-based access to the operating system, in contrast to the mostly graphical nature of the user experience of macOS, by providing a [command line interface](#) to the operating system