

# Cellular Automata

Weston Dransfield

April 1, 2016

# Outline

# The Basics

- ▶ Spatial Structure

# The Basics

- ▶ Spatial Structure
- ▶ Local Interactions

# The Basics

- ▶ Spatial Structure
- ▶ Local Interactions
- ▶ Cell State

# The Basics

- ▶ Spatial Structure
- ▶ Local Interactions
- ▶ Cell State
- ▶ Cell Transitions

# Spatial Structure

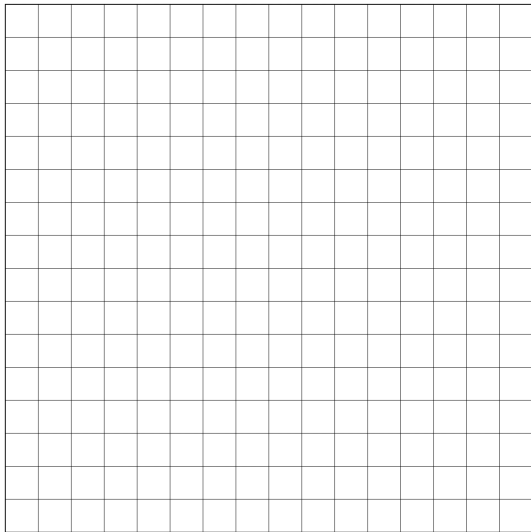
- ▶ Grid - Like a city

# Spatial Structure

- ▶ Grid - Like a city
- ▶ Agents - Like a home in the city



# Spatial Structure



papergraph.com © 3.0 by-nc-sa Pere Millán

# Local Interactions

Agents may only interact with a set number of agents around them. This region is called a neighborhood.

- ▶ Von Neumann neighborhood (most local).

# Local Interactions

Agents may only interact with a set number of agents around them. This region is called a neighborhood.

- ▶ Von Neumann neighborhood (most local).
- ▶ Moore neighborhood (less local).

# Local Interactions

Agents may only interact with a set number of agents around them. This region is called a neighborhood.

- ▶ Von Neumann neighborhood (most local).
- ▶ Moore neighborhood (less local).
- ▶  $n * n$  Moore neighborhood for  $n > 3$  (least local).

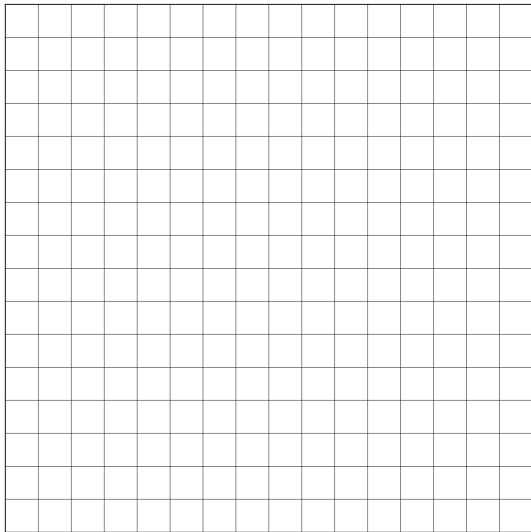
# Local Interactions

Agents may only interact with a set number of agents around them. This region is called a neighborhood.

- ▶ Von Neumann neighborhood (most local).
- ▶ Moore neighborhood (less local).
- ▶  $n * n$  Moore neighborhood for  $n > 3$  (least local).

Locality indicates how many agents a single agent interacts with. High locality indicates more interactions (Think of how a rumor spreads).

# Local Interactions



papergraph.com © 3.0 by-nc-sa Pere Millán

# Cell State

Cells have a state

- ▶ Binary

# Cell State

Cells have a state

- ▶ Binary
- ▶ Any number of states



# Cell Transitions

CA have discrete chunks of time called rounds. Each round produces the next "generation" of cells.

# Cell Transitions

The next state of a cell is a function of:

- ▶ It's current state

# Cell Transitions

The next state of a cell is a function of:

- ▶ It's current state
- ▶ The state of cells in its neighborhood

# Cell Transitions

The next state of a cell is a function of:

- ▶ It's current state
- ▶ The state of cells in its neighborhood

Think of your opinion on a topic changing by being surrounded by people with a different view.

# Elementary Cellular Automaton

What is the simplest...

- ▶ Grid of cells?

# Elementary Cellular Automaton

What is the simplest...

- ▶ Grid of cells?
- ▶ Useful number of states?

# Elementary Cellular Automaton

What is the simplest...

- ▶ Grid of cells?
- ▶ Useful number of states?
- ▶ Neighborhood?

# Elementary Cellular Automaton

What is the simplest...

- ▶ Grid of cells - an array
- ▶ Useful number of states - binary
- ▶ Neighborhood - 2 adjacent cells of the agent



# Elementary Cellular Automaton (Exercise)

How do we describe the state of cell  $c_t$  as a function of the neighbors of  $c_t$  and the state of  $c_t$  at time  $t - 1$ ?

# Elementary Cellular Automaton (Exercise)

How do we describe the state of cell  $c_t$  as a function of the neighbors of  $c_t$  and the state of  $c_t$  at time  $t - 1$ ?

- ▶ There are  $2^3 = 8$  neighborhood combinations for any given neighborhood.

# Elementary Cellular Automaton (Exercise)

How do we describe the state of cell  $c_t$  as a function of the neighbors of  $c_t$  and the state of  $c_t$  at time  $t - 1$ ?

- ▶ There are  $2^3 = 8$  neighborhood combinations for any given neighborhood.
- ▶ Simply create an assignment for each combination!

# Elementary Cellular Automaton (Exercise)

How do we describe the state of cell  $c_t$  as a function of the neighbors of  $c_t$  and the state of  $c_t$  at time  $t - 1$ ?

- ▶ There are  $2^3 = 8$  neighborhood combinations for any given neighborhood.
- ▶ Simply create an assignment for each combination!
- ▶ This assignment is called a rule set (how many possible rule sets?)

# Example Program

- ▶ `http://mathworld.wolfram.com/ElementaryCellularAutomaton.html`
- ▶ `https://github.com/westonkd/Completeness/tree/CA`

# Interesting Tidbits

- ▶ Rule 30 - Random number generator for Wolfram language

# Interesting Tidbits

- ▶ Rule 30 - Random number generator for Wolfram language
- ▶ Rule 22 - Sierpinski triangle

## 2D CA Example - Conway's Game of Life





# Conway's Game of Life

## 1. Rules

# Conway's Game of Life

1. Rules
2. Entities

# Conway's Game of Life

1. Rules
2. Entities
3. Results

# Conway's Game of Life

1. Rules
2. Entities
3. Results
4. Computational tidbits

# Conway's Game of Life Rules

- ▶ 2 states - dead or alive

# Conway's Game of Life Rules

- ▶ 2 states - dead or alive
- ▶ 3x3 Moore neighborhood

# Conway's Game of Life Rules

- ▶ 2 states - dead or alive
  - ▶ 3x3 Moore neighborhood
1. Any live cell with fewer than two live neighbors dies (under population).

# Conway's Game of Life Rules

- ▶ 2 states - dead or alive
  - ▶ 3x3 Moore neighborhood
1. Any live cell with fewer than two live neighbors dies (under population).
  2. Any cell with 2 or 3 live neighbors lives on to the next generation.



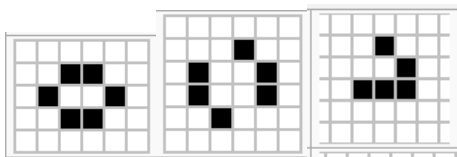
# Conway's Game of Life Rules

- ▶ 2 states - dead or alive
  - ▶ 3x3 Moore neighborhood
1. Any live cell with fewer than two live neighbors dies (under population).
  2. Any cell with 2 or 3 live neighbors lives on to the next generation.
  3. A cell with more than 3 live neighbors dies (over crowding).

# Conway's Game of Life Rules

- ▶ 2 states - dead or alive
  - ▶ 3x3 Moore neighborhood
1. Any live cell with fewer than two live neighbors dies (under population).
  2. Any cell with 2 or 3 live neighbors lives on to the next generation.
  3. A cell with more than 3 live neighbors dies (over crowding).
  4. A dead cell with exactly 3 live neighbors becomes alive (reproduction).

# Practice



# Conway's Game of Life Entities

1. Still lives - entity stays the same through generations

https:  
[//en.wikipedia.org/wiki/Conway%27s\\_Game\\_of\\_Life](https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life)

# Conway's Game of Life Entities

1. Still lives - entity stays the same through generations
2. Oscillators - entity changes shape and returns to original position (periods)

https:  
[//en.wikipedia.org/wiki/Conway%27s\\_Game\\_of\\_Life](https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life)

# Conway's Game of Life Entities

1. Still lives - entity stays the same through generations
2. Oscillators - entity changes shape and returns to original position (periods)
3. Spaceships - Moving oscillators

https:

[//en.wikipedia.org/wiki/Conway%27s\\_Game\\_of\\_Life](https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life)

# Conway's Game of Life Results

1. Fade away completely

# Conway's Game of Life Results

1. Fade away completely
2. Stable configuration



# Conway's Game of Life Results

1. Fade away completely
2. Stable configuration
3. Oscillating phase

# Conway's Game of Life Results

1. Fade away completely
2. Stable configuration
3. Oscillating phase
4. ...Gosper glider gun

# Demo

- ▶ <http://www.bitstorm.org/gameoflife/>

# Conway's Game of Life Turing Completeness

- ▶ Conway's Game of Life is Turing Complete
- ▶ <http://rendell-attic.org/gol/tm.htm>
- ▶ <https://www.youtube.com/watch?v=My8AsV7bA94>

# Conway's Game of Life Decidability

The question - "Given a starting pattern and an ending pattern, will the starting pattern ever reach the ending pattern?"

# Conway's Game of Life Decidability

The question - "Given a starting pattern and an ending pattern, will the starting pattern ever reach the ending pattern?"

- ▶ Undecidable - halting problem

# Conway's Game of Life Decidability

The question - "Given a starting pattern and an ending pattern, will the starting pattern ever reach the ending pattern?"

- ▶ Undecidable - halting problem
- ▶ "Indeed, since the game of life includes a pattern that is equivalent to a UTM (universal Turing machine), this 'deciding' algorithm, if existed, could have been used to solve the halting problem, by taking the initial pattern as the one corresponding to a UTM+input and the later pattern as the one corresponding to a halting state of the machine with an empty tape (as one can modify the Turing machine to always erase the tape before halting). However the halting problem is provably undecidable and so such an algorithm does not exist" (Wikipedia).

# CAs in the Wild

- ▶ Image processing (pixels)



# CAs in the Wild

- ▶ Image processing (pixels)
- ▶ Nature

# CAs in the Wild

- ▶ Image processing (pixels)
- ▶ Nature
- ▶ Migration patterns

## CAs in the Wild



## Extra: 3D CAs

`http://cubes.io/`

# Going Further

- ▶ Crazy grids

# Going Further

- ▶ Crazy grids
- ▶ Probability

# Going Further

- ▶ Crazy grids
- ▶ Probability
- ▶ History

# Sources

- ▶ [https://en.wikipedia.org/wiki/Conway%27s\\_Game\\_of\\_Life](https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life)
- ▶ <https://www.youtube.com/watch?v=W1zKu3fDQR8>
- ▶ <https://www.youtube.com/watch?v=Eyrw0f239M4>
- ▶ [https://www3.nd.edu/~mtns/papers/17761\\_4.pdf](https://www3.nd.edu/~mtns/papers/17761_4.pdf)
- ▶ <http://www.sciencedirect.com/science/article/pii/S089571771000333X>