Cellular automata and agent-based models

Matthew Macauley

Department of Mathematical Sciences
Clemson University
http://www.math.clemson.edu/~macaule/

Math 4500, Fall 2016

Cellular automata

Cellular automata

A cellular automaton (CA) consists of a regular grid of cells, each one being ON (1) or OFF (0).

Cellular automata

A cellular automaton (CA) consists of a regular grid of cells, each one being ON (1) or OFF (0).

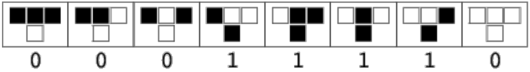
At each time-step, every state is updated based on the states of its neighbors.

Cellular automata

A cellular automaton (CA) consists of a regular grid of cells, each one being ON (1) or OFF (0).

At each time-step, every state is updated based on the states of its neighbors.

As a simple example, consider an infinite 1D grid of cells, each one having the following update rule, called "Rule 30":

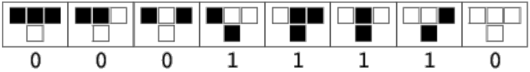


Cellular automata

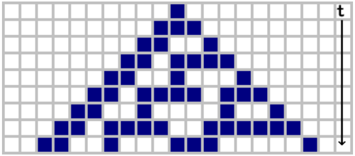
A cellular automaton (CA) consists of a regular grid of cells, each one being ON (1) or OFF (0).

At each time-step, every state is updated based on the states of its neighbors.

As a simple example, consider an infinite 1D grid of cells, each one having the following update rule, called "Rule 30":



The following shows the evolution of the dynamics over $t = 0, 1, \ldots, 8$, starting with a single "ON" cell:

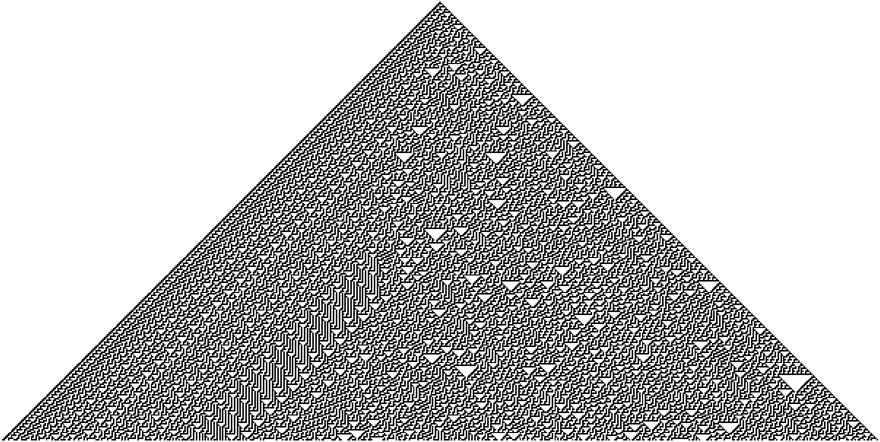


Cellular automata

When you zoom out to see 200 time-steps, patterns start to emerge.

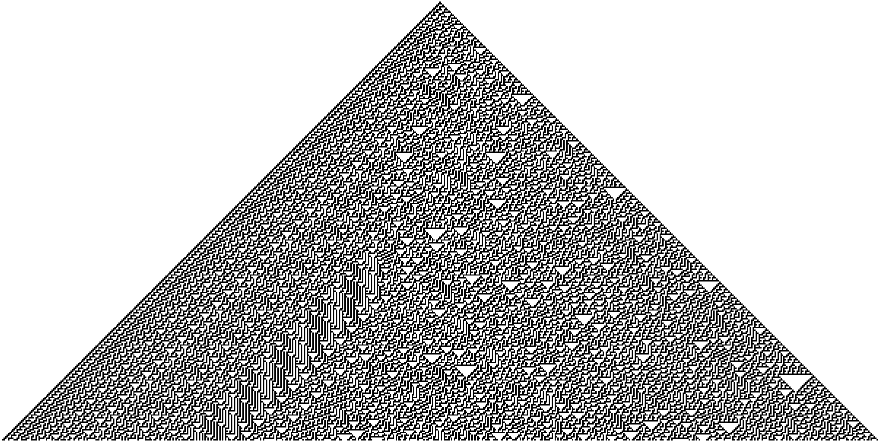
Cellular automata

When you zoom out to see 200 time-steps, patterns start to emerge.



Cellular automata

When you zoom out to see 200 time-steps, patterns start to emerge.



A common theme with CA are that complex dynamics can emerge from simple, local interactions.

Cellular automata and self-organizing systems

Cellular automata and self-organizing systems

Complexity is observed all throughout the natural world, expecially in biology.

Cellular automata and self-organizing systems

Complexity is observed all throughout the natural world, expecially in biology.

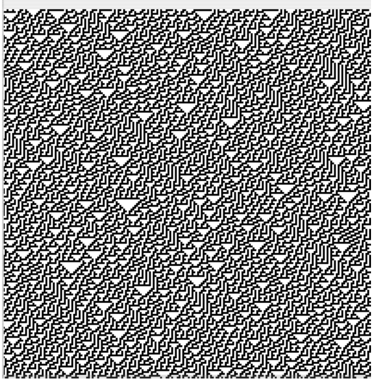
Question: *Can complex bevavior emerge naturally from a few simple rules?*

Cellular automata and self-organizing systems

Complexity is observed all throughout the natural world, expecially in biology.

Question: *Can complex bevavior emerge naturally from a few simple rules?*

YES! For example, here is Rule 30 with a different initial condition:

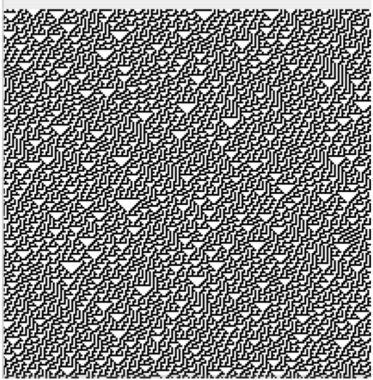


Cellular automata and self-organizing systems

Complexity is observed all throughout the natural world, expecially in biology.

Question: *Can complex bevavior emerge naturally from a few simple rules?*

YES! For example, here is Rule 30 with a different initial condition:



Many believe that CAs are key to understanding how simple rules can produce complex structures and behavior.

Some history

Cellular automata (CA) were invented by Stanislaw Ulam and John von Neumann in the 1940s at Los Alamos National Laboratory, based on work by Alan Turing.

Some history

Cellular automata (CA) were invented by Stanislaw Ulam and John von Neumann in the 1940s at Los Alamos National Laboratory, based on work by Alan Turing.

Three major events in CA research

Some history

Cellular automata (CA) were invented by Stanislaw Ulam and John von Neumann in the 1940s at Los Alamos National Laboratory, based on work by Alan Turing.

Three major events in CA research

- John von Neumann's self-reproducing automaton (1940s)

Some history

Cellular automata (CA) were invented by Stanislaw Ulam and John von Neumann in the 1940s at Los Alamos National Laboratory, based on work by Alan Turing.

Three major events in CA research

- John von Neumann's self-reproducing automaton (1940s)
- John Conway's Game of Life (1970s)

Some history

Cellular automata (CA) were invented by Stanislaw Ulam and John von Neumann in the 1940s at Los Alamos National Laboratory, based on work by Alan Turing.

Three major events in CA research

- John von Neumann's self-reproducing automaton (1940s)
- John Conway's Game of Life (1970s)
- Stephen Wolfram's classification of cellular automata (1980s)

Some history

Cellular automata (CA) were invented by Stanislaw Ulam and John von Neumann in the 1940s at Los Alamos National Laboratory, based on work by Alan Turing.

Three major events in CA research

- John von Neumann's self-reproducing automaton (1940s)
- John Conway's Game of Life (1970s)
- Stephen Wolfram's classification of cellular automata (1980s)

CA were popularized in the 1970s by John Conway's Game of Life (click here!)

Some history

Cellular automata (CA) were invented by Stanislaw Ulam and John von Neumann in the 1940s at Los Alamos National Laboratory, based on work by Alan Turing.

Three major events in CA research

- John von Neumann's self-reproducing automaton (1940s)
- John Conway's Game of Life (1970s)
- Stephen Wolfram's classification of cellular automata (1980s)

CA were popularized in the 1970s by John Conway's Game of Life (click here!)

Game of Life rules

Some history

Cellular automata (CA) were invented by Stanislaw Ulam and John von Neumann in the 1940s at Los Alamos National Laboratory, based on work by Alan Turing.

Three major events in CA research

- John von Neumann's self-reproducing automaton (1940s)
- John Conway's Game of Life (1970s)
- Stephen Wolfram's classification of cellular automata (1980s)

CA were popularized in the 1970s by John Conway's Game of Life (click here!)

Game of Life rules

- Infinite 2-dimensional square grid of cells.

Some history

Cellular automata (CA) were invented by Stanislaw Ulam and John von Neumann in the 1940s at Los Alamos National Laboratory, based on work by Alan Turing.

Three major events in CA research

- John von Neumann's self-reproducing automaton (1940s)
- John Conway's Game of Life (1970s)
- Stephen Wolfram's classification of cellular automata (1980s)

CA were popularized in the 1970s by John Conway's Game of Life (click here!)

Game of Life rules

- Infinite 2-dimensional square grid of cells.
- Each cell is either ALIVE (i.e., "on" or 1) or DEAD (i.e., "off" or 0).

Some history

Cellular automata (CA) were invented by Stanislaw Ulam and John von Neumann in the 1940s at Los Alamos National Laboratory, based on work by Alan Turing.

Three major events in CA research

- John von Neumann's self-reproducing automaton (1940s)
- John Conway's Game of Life (1970s)
- Stephen Wolfram's classification of cellular automata (1980s)

CA were popularized in the 1970s by John Conway's Game of Life (click here!)

Game of Life rules

- Infinite 2-dimensional square grid of cells.
- Each cell is either ALIVE (i.e., "on" or 1) or DEAD (i.e., "off" or 0).
- The state of each cell is determined by the states of its 8 neighbors.

Some history

Cellular automata (CA) were invented by Stanislaw Ulam and John von Neumann in the 1940s at Los Alamos National Laboratory, based on work by Alan Turing.

Three major events in CA research

- John von Neumann's self-reproducing automaton (1940s)
- John Conway's Game of Life (1970s)
- Stephen Wolfram's classification of cellular automata (1980s)

CA were popularized in the 1970s by John Conway's Game of Life (click here!)

Game of Life rules

- Infinite 2-dimensional square grid of cells.
- Each cell is either ALIVE (i.e., "on" or 1) or DEAD (i.e., "off" or 0).
- The state of each cell is determined by the states of its 8 neighbors.
- Update rules:
  - If a cell is alive, then it stays alive iff it has either 2 or 3 live neighbors.

Some history

Cellular automata (CA) were invented by Stanislaw Ulam and John von Neumann in the 1940s at Los Alamos National Laboratory, based on work by Alan Turing.

Three major events in CA research

- John von Neumann's self-reproducing automaton (1940s)
- John Conway's Game of Life (1970s)
- Stephen Wolfram's classification of cellular automata (1980s)

CA were popularized in the 1970s by John Conway's Game of Life (click here!)

Game of Life rules

- Infinite 2-dimensional square grid of cells.
- Each cell is either ALIVE (i.e., "on" or 1) or DEAD (i.e., "off" or 0).
- The state of each cell is determined by the states of its 8 neighbors.
- Update rules:
  - If a cell is alive, then it stays alive iff it has either 2 or 3 live neighbors.
  - If a cell is dead, then it becomes alive iff it has exactly live 3 neighbors.

Some history

Cellular automata (CA) were invented by Stanislaw Ulam and John von Neumann in the 1940s at Los Alamos National Laboratory, based on work by Alan Turing.

Three major events in CA research

- John von Neumann's self-reproducing automaton (1940s)
- John Conway's Game of Life (1970s)
- Stephen Wolfram's classification of cellular automata (1980s)

CA were popularized in the 1970s by John Conway's Game of Life (click here!)

Game of Life rules

- Infinite 2-dimensional square grid of cells.
- Each cell is either ALIVE (i.e., "on" or 1) or DEAD (i.e., "off" or 0).
- The state of each cell is determined by the states of its 8 neighbors.
- Update rules:
  - If a cell is alive, then it stays alive iff it has either 2 or 3 live neighbors.
  - If a cell is dead, then it becomes alive iff it has exactly live 3 neighbors.

There are many variations, leading to an endless potential of research projects.

Elementary cellular automata

Elementary cellular automata

Stephen Wolfram (born 1959) is a British physicist who got his PhD at Caltech at the age of 20, and joined the faculty the following year.

Elementary cellular automata

Stephen Wolfram (born 1959) is a British physicist who got his PhD at Caltech at the age of 20, and joined the faculty the following year.

He co-founded Wolfram Research, which developed *Mathematica* and *WolframAlpha*.

Elementary cellular automata

Stephen Wolfram (born 1959) is a British physicist who got his PhD at Caltech at the age of 20, and joined the faculty the following year.

He co-founded Wolfram Research, which developed *Mathematica* and *WolframAlpha*.

In 2002, Wolfram published a best-selling 1197-page book *A New Kind of Science*, promoting Cellular automata as a vital aspect of modern science.

Elementary cellular automata

Stephen Wolfram (born 1959) is a British physicist who got his PhD at Caltech at the age of 20, and joined the faculty the following year.

He co-founded Wolfram Research, which developed *Mathematica* and *WolframAlpha*.

In 2002, Wolfram published a best-selling 1197-page book *A New Kind of Science*, promoting Cellular automata as a vital aspect of modern science.

Wolfram advocated studying the $2^8 = 256$ one-dimensional CA rules, called *Elementary cellular automata* (ECA).

Elementary cellular automata

Stephen Wolfram (born 1959) is a British physicist who got his PhD at Caltech at the age of 20, and joined the faculty the following year.

He co-founded Wolfram Research, which developed *Mathematica* and *WolframAlpha*.

In 2002, Wolfram published a best-selling 1197-page book *A New Kind of Science*, promoting Cellular automata as a vital aspect of modern science.

Wolfram advocated studying the $2^8 = 256$ one-dimensional CA rules, called *Elementary cellular automata* (ECA).

The numbering scheme is done by translating the function output from a binary string into a decimal number from 0–255.

Elementary cellular automata

Stephen Wolfram (born 1959) is a British physicist who got his PhD at Caltech at the age of 20, and joined the faculty the following year.

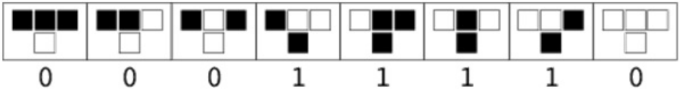
He co-founded Wolfram Research, which developed *Mathematica* and *WolframAlpha*.

In 2002, Wolfram published a best-selling 1197-page book *A New Kind of Science*, promoting Cellular automata as a vital aspect of modern science.

Wolfram advocated studying the $2^8 = 256$ one-dimensional CA rules, called *Elementary cellular automata* (ECA).

The numbering scheme is done by translating the function output from a binary string into a decimal number from 0–255.

For example, what ECA rule is this?



Wolfram's classification of ECA rules

Elementary cellular automata

Wolfram's classification of ECA rules

The 256 ECA rules fall into 4 "classes" based on their dynamical complexity.

Elementary cellular automata

Wolfram's classification of ECA rules

The 256 ECA rules fall into 4 "classes" based on their dynamical complexity.

- Class 1: Converge quickly to a uniform state

Elementary cellular automata

Wolfram's classification of ECA rules

The 256 ECA rules fall into 4 "classes" based on their dynamical complexity.

- Class 1: Converge quickly to a uniform state
- Class 2: Converge quickly to stable or periodic structures

Elementary cellular automata

Wolfram's classification of ECA rules

The 256 ECA rules fall into 4 "classes" based on their dynamical complexity.

- Class 1: Converge quickly to a uniform state
- Class 2: Converge quickly to stable or periodic structures
- Class 3: Chaotic aperiodic behavior

Elementary cellular automata

Wolfram's classification of ECA rules

The 256 ECA rules fall into 4 "classes" based on their dynamical complexity.

- Class 1: Converge quickly to a uniform state
- Class 2: Converge quickly to stable or periodic structures
- Class 3: Chaotic aperiodic behavior
- Class 4: Complicated localized structures

Elementary cellular automata

Wolfram's classification of ECA rules

The 256 ECA rules fall into 4 "classes" based on their dynamical complexity.

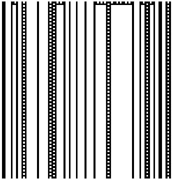
- Class 1: Converge quickly to a uniform state
- Class 2: Converge quickly to stable or periodic structures
- Class 3: Chaotic aperiodic behavior
- Class 4: Complicated localized structures

Elementary cellular automata

Wolfram's classification of ECA rules

The 256 ECA rules fall into 4 "classes" based on their dynamical complexity.

- Class 1: Converge quickly to a uniform state
- Class 2: Converge quickly to stable or periodic structures
- Class 3: Chaotic aperiodic behavior
- Class 4: Complicated localized structures

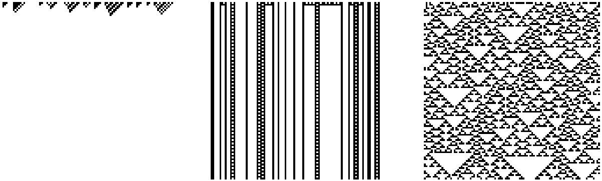


Elementary cellular automata

Wolfram's classification of ECA rules

The 256 ECA rules fall into 4 "classes" based on their dynamical complexity.

- Class 1: Converge quickly to a uniform state
- Class 2: Converge quickly to stable or periodic structures
- Class 3: Chaotic aperiodic behavior
- Class 4: Complicated localized structures



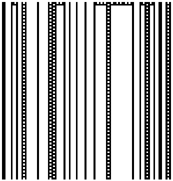
Elementary cellular automata

Wolfram's classification of ECA rules

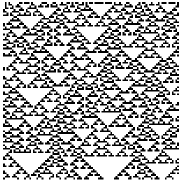
The 256 ECA rules fall into 4 "classes" based on their dynamical complexity.

- Class 1: Converge quickly to a uniform state
- Class 2: Converge quickly to stable or periodic structures
- Class 3: Chaotic aperiodic behavior
- Class 4: Complicated localized structures

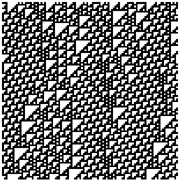
Rule 168



Rule 108



Rule 22



Rule 110

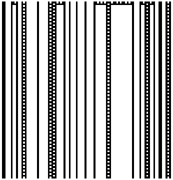
Elementary cellular automata

Wolfram's classification of ECA rules

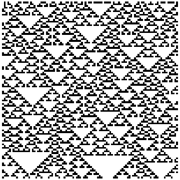
The 256 ECA rules fall into 4 "classes" based on their dynamical complexity.

- Class 1: Converge quickly to a uniform state
- Class 2: Converge quickly to stable or periodic structures
- Class 3: Chaotic aperiodic behavior
- Class 4: Complicated localized structures

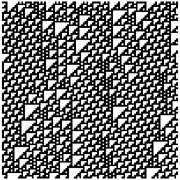
Rule 168



Rule 108



Rule 22



Rule 110

Problem

Class membership of a given rule is computationally *undecidable*!

Some interesting ECA rules

Some interesting ECA rules

Rule 30

Some interesting ECA rules

Rule 30

- Displays chaotic, aperiodic behavior.

Some interesting ECA rules

Rule 30

- Displays chaotic, aperiodic behavior.
- Similar patterns appear on the shell of the *Conus textile* snail.

Some interesting ECA rules

Rule 30

- Displays chaotic, aperiodic behavior.
- Similar patterns appear on the shell of the *Conus textile* snail.
- Has been used as a random number generator in Mathematica.

Some interesting ECA rules

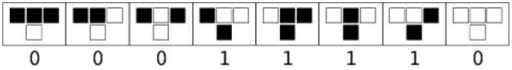
Rule 30

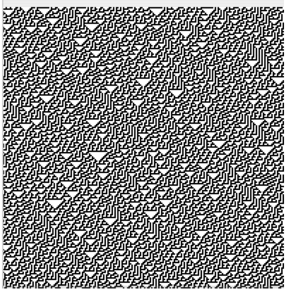
- Displays chaotic, aperiodic behavior.
- Similar patterns appear on the shell of the *Conus textile* snail.
- Has been used as a random number generator in Mathematica.
- Has been proposed as a possible stream cipher in cryptography.

Some interesting ECA rules

Rule 30

- Displays chaotic, aperiodic behavior.
- Similar patterns appear on the shell of the *Conus textile* snail.
- Has been used as a random number generator in Mathematica.
- Has been proposed as a possible stream cipher in cryptography.





Some interesting ECA rules

Rule 184

Some interesting ECA rules

Rule 184

- Used as a simple model of traffic flow in a single lane.

Some interesting ECA rules

Rule 184

- Used as a simple model of traffic flow in a single lane.
- Used as a model of surface deposition (particle physics).

Some interesting ECA rules

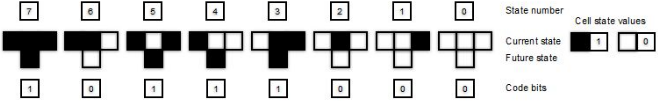
Rule 184

- Used as a simple model of traffic flow in a single lane.
- Used as a model of surface deposition (particle physics).
- Used as a model of ballistic annihilation (particle physics).

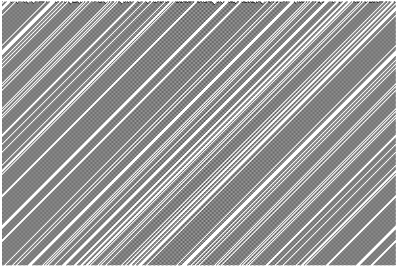
Some interesting ECA rules

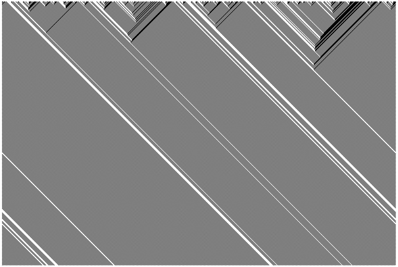
Rule 184

- Used as a simple model of traffic flow in a single lane.
- Used as a model of surface deposition (particle physics).
- Used as a model of ballistic annihilation (particle physics).



Rule 184





Some interesting ECA rules

Rule 90

Some interesting ECA rules

Rule 90

- XOR function; has been called the "simplest non-trivial cellular automaton".

Some interesting ECA rules

Rule 90

- XOR function; has been called the "simplest non-trivial cellular automaton".
- Discussed extensively in *A New Kind of Science*.

Some interesting ECA rules

Rule 90

- XOR function; has been called the "simplest non-trivial cellular automaton".
- Discussed extensively in *A New Kind of Science*.
- Random initial configurations remain random at each time-step.

Some interesting ECA rules

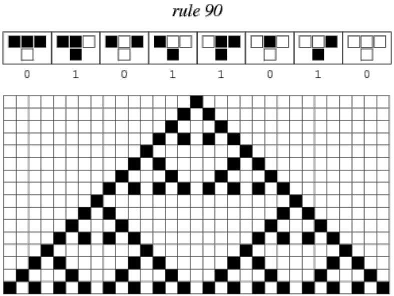
Rule 90

- XOR function; has been called the "simplest non-trivial cellular automaton".
- Discussed extensively in *A New Kind of Science*.
- Random initial configurations remain random at each time-step.
- Any finite configuration becomes a "replicator", filling all cells with copies of itself.

Some interesting ECA rules

Rule 90

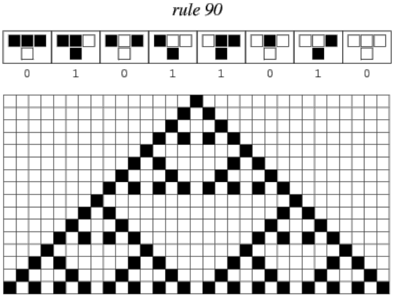
- XOR function; has been called the "simplest non-trivial cellular automaton".
- Discussed extensively in *A New Kind of Science*.
- Random initial configurations remain random at each time-step.
- Any finite configuration becomes a "replicator", filling all cells with copies of itself.

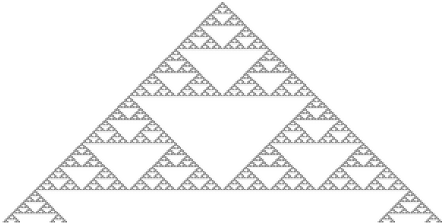


Some interesting ECA rules

Rule 90

- XOR function; has been called the "simplest non-trivial cellular automaton".
- Discussed extensively in *A New Kind of Science*.
- Random initial configurations remain random at each time-step.
- Any finite configuration becomes a "replicator", filling all cells with copies of itself.





Some interesting ECA rules

Rule 110

Some interesting ECA rules

Rule 110

- Lies on the boundary between order and chaos.

Some interesting ECA rules

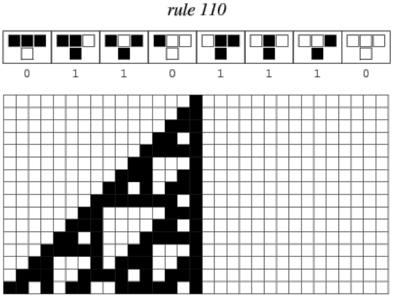
Rule 110

- Lies on the boundary between order and chaos.
- In the late 1990s, Matthew Cook proved that Rule 110 is *Turing complete*, which means that it can simulate the logic of any computer algorithm.

Some interesting ECA rules

Rule 110

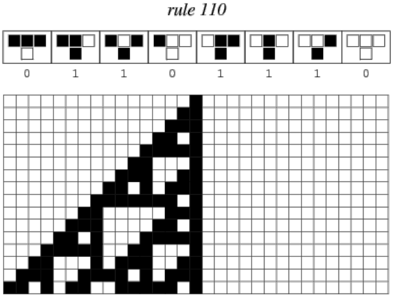
- Lies on the boundary between order and chaos.
- In the late 1990s, Matthew Cook proved that Rule 110 is *Turing complete*, which means that it can simulate the logic of any computer algorithm.

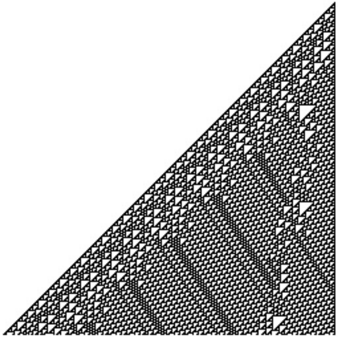


Some interesting ECA rules

Rule 110

- Lies on the boundary between order and chaos.
- In the late 1990s, Matthew Cook proved that Rule 110 is *Turing complete*, which means that it can simulate the logic of any computer algorithm.

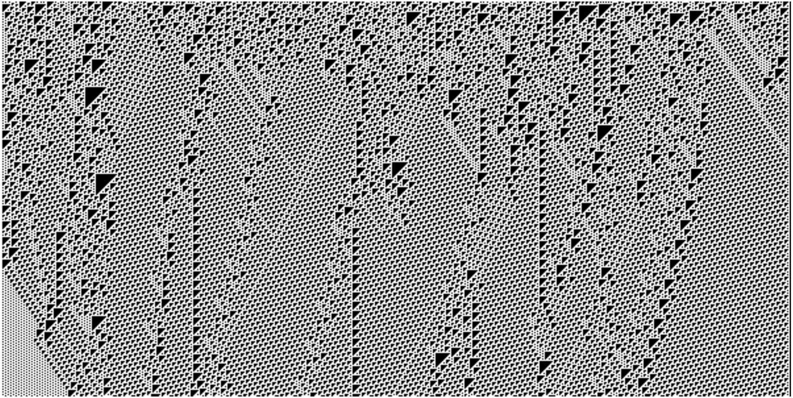




Some interesting ECA rules

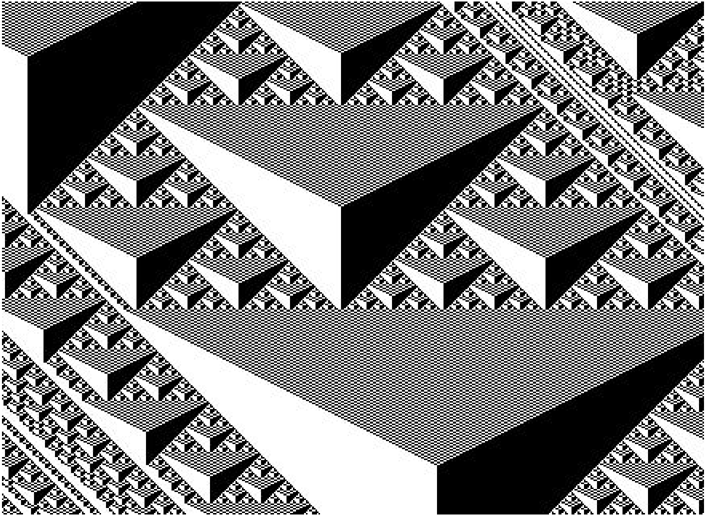
Rule 110

- Lies on the boundary between order and chaos.
- In the last 1990s, Matthew Cook proved that Rule 110 was *Turing complete*, which means that it can simulate the logic of any computer algorithm.



Generalizations of elementary cellular automata

There are $2^{2^5} = 4294967296$ one-dimensional "2-neighbor rules". Here is one of them:



Why study cellular automata, or the Game of Life?

Why study cellular automata, or the Game of Life?

- Studying the patterns in Life can lead to discoveries in other areas of math and science.

Why study cellular automata, or the Game of Life?

- Studying the patterns in Life can lead to discoveries in other areas of math and science.

- The behavior of cells and/or animals can be better understood using simple rules. Behaviors that seem intelligent (ant or bee colonies, flocks of birds, schools of fish, etc.) might just be simple rules that we don't understand.

Why study cellular automata, or the Game of Life?

- Studying the patterns in Life can lead to discoveries in other areas of math and science.

- The behavior of cells and/or animals can be better understood using simple rules. Behaviors that seem intelligent (ant or bee colonies, flocks of birds, schools of fish, etc.) might just be simple rules that we don't understand.

- Good way to model traffic flow.

Why study cellular automata, or the Game of Life?

- Studying the patterns in Life can lead to discoveries in other areas of math and science.
- The behavior of cells and/or animals can be better understood using simple rules. Behaviors that seem intelligent (ant or bee colonies, flocks of birds, schools of fish, etc.) might just be simple rules that we don't understand.
- Good way to model traffic flow.
- Computer viruses can be viewed as cellular automata. Cures could be hidden in the patterns of this simple game.

Why study cellular automata, or the Game of Life?

- Studying the patterns in Life can lead to discoveries in other areas of math and science.
- The behavior of cells and/or animals can be better understood using simple rules. Behaviors that seem intelligent (ant or bee colonies, flocks of birds, schools of fish, etc.) might just be simple rules that we don't understand.
- Good way to model traffic flow.
- Computer viruses can be viewed as cellular automata. Cures could be hidden in the patterns of this simple game.
- Understaning why cells live and die could have a profound effect on medicine.

Why study cellular automata, or the Game of Life?

- Studying the patterns in Life can lead to discoveries in other areas of math and science.
- The behavior of cells and/or animals can be better understood using simple rules. Behaviors that seem intelligent (ant or bee colonies, flocks of birds, schools of fish, etc.) might just be simple rules that we don't understand.
- Good way to model traffic flow.
- Computer viruses can be viewed as cellular automata. Cures could be hidden in the patterns of this simple game.
- Understaning why cells live and die could have a profound effect on medicine.
- Exploring the universe would be easier if machines could be invented that could build themselves. Imagine sending a probe to Mars that could build a copy of itself.

Other agent-based models

Other agent-based models

Cellular automata promote the idea that complex dynamics can arise from simple local rules.

Agent based model of flocking

Agent based model of flocking

An *agent-based model* (ABM) is a type of computational model for simulating the behavior of interacting autonomous individuals, called "agents".

Agent based model of flocking

An *agent-based model* (ABM) is a type of computational model for simulating the behavior of interacting autonomous individuals, called "agents".

For example, consider a flock of a few thousand birds that fly around "randomly", each following three rules: "alignment", "separation", and "cohesion".

Agent based model of flocking

An *agent-based model* (ABM) is a type of computational model for simulating the behavior of interacting autonomous individuals, called "agents".

For example, consider a flock of a few thousand birds that fly around "randomly", each following three rules: "alignment", "separation", and "cohesion".

- alignment: a bird tends to fly in the same direction as its neighbors

Agent based model of flocking

An *agent-based model* (ABM) is a type of computational model for simulating the behavior of interacting autonomous individuals, called "agents".

For example, consider a flock of a few thousand birds that fly around "randomly", each following three rules: "alignment", "separation", and "cohesion".

- alignment: a bird tends to fly in the same direction as its neighbors
- separation: a bird will turn to avoid another bird that gets too close

Agent based model of flocking

An *agent-based model* (ABM) is a type of computational model for simulating the behavior of interacting autonomous individuals, called "agents".

For example, consider a flock of a few thousand birds that fly around "randomly", each following three rules: "alignment", "separation", and "cohesion".

- alignment: a bird tends to fly in the same direction as its neighbors
- separation: a bird will turn to avoid another bird that gets too close
- cohesion: a bird will move towards nearby birds

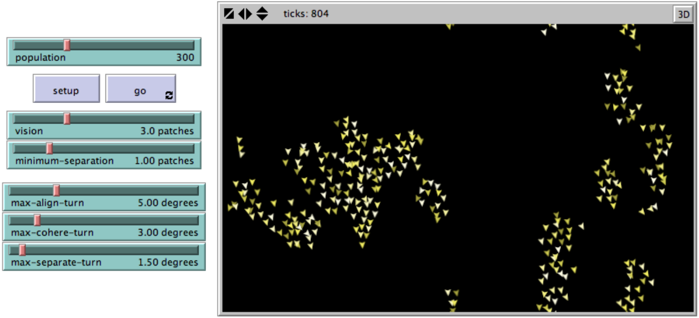
Agent based model of flocking

An *agent-based model* (ABM) is a type of computational model for simulating the behavior of interacting autonomous individuals, called "agents".

For example, consider a flock of a few thousand birds that fly around "randomly", each following three rules: "alignment", "separation", and "cohesion".

- alignment: a bird tends to fly in the same direction as its neighbors
- separation: a bird will turn to avoid another bird that gets too close
- cohesion: a bird will move towards nearby birds

NetLogo is a wonderful free open-source program that has many built-in agent-based models such as this one.



Agent based model of segregation

Agent based model of segregation

Consider a population of tigers; half are orange and half are purple.

Agent based model of segregation

Consider a population of tigers; half are orange and half are purple.

Suppose that each tiger has a slight subconscience preference of having neighbors of the same color.

Agent based model of segregation

Consider a population of tigers; half are orange and half are purple.

Suppose that each tiger has a slight subconscience preference of having neighbors of the same color.

If less than 33% of a tiger's neighbors are of the same color, it will continue looking for a suitable place to live.

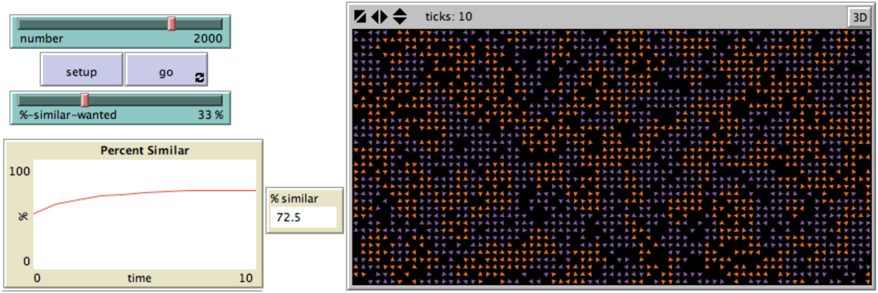
Agent based model of segregation

Consider a population of tigers; half are orange and half are purple.

Suppose that each tiger has a slight subconscience preference of having neighbors of the same color.

If less than 33% of a tiger's neighbors are of the same color, it will continue looking for a suitable place to live.

Here is a result of a NetLogo simulation showing where the tiger end up living.



Other agent-based models in NetLogo

Other agent-based models in NetLogo

NetLogo has hundreds of built-in agent-based models, such as:

- firefly synchronization
- voting communities
- percolation (oil spill through soil)
- traffic grids
- forest fires
- moths and light
- enzyme kinetics
- tumor growth
- virus spreading through a network
- shepherds herding sheep
- predator-prey
- ⋮