UNIT 9C
Randomness in Computation: Cellular Automata

Announcements

- Exam locations:

  2:30 Exam: Sections A, B, C, D, E go to Rashid (GHC 4401)

  Sections F, G go to PH 125C.

  3:30 Exam: All sections go to Rashid (GHC 4401).

- Review sessions
  - Sunday 6-8 and 8-10 at GHC 4303
- Office hours on Monday and Tuesday
  - In PHA 18C, not in clusters

Last Two Lectures

- A computer is deterministic. It follows rules, step by step. Then, can we get it to behave randomly?
  - Linear congruential method for generating "seemingly" random integers
  - Based on such a method we can implement higher level functions to give us "random" outcomes

This Lecture

- Another example of how complex systems can be generated by very simple programs
  - Relevance to the unit: Simple rules to generate seemingly random structures

Cellular Automata

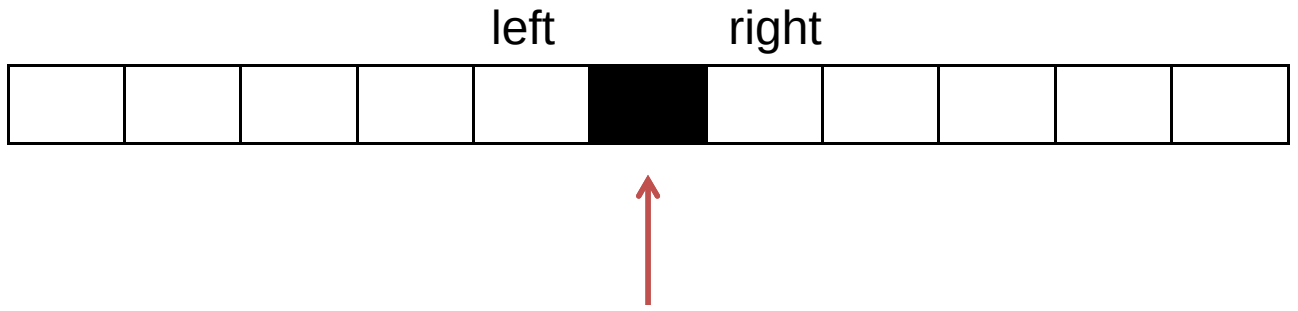
- A cellular automaton is a collection of cells on a grid that evolves through a number of discrete time steps (generations) according to a set of rules based on the states of neighboring cells.
- The rules are then applied iteratively for as many time steps as desired.
  - John von Neumann was one of the first people to consider such a model.

(from Wolfram MathWorld)

One-Dimensional Cellular Automata

- Every cell has with two possible states indicated by black or white

- A cell's neighbors defined to be the adjacent cells on either side of it.



black cell with a white left neighbor and a white right neighbor

Evolution of the Automaton

Starting from an initial configuration a new generation is obtained by applying a given rule to find the state of each cell in the next generation

generation 0

generation 1

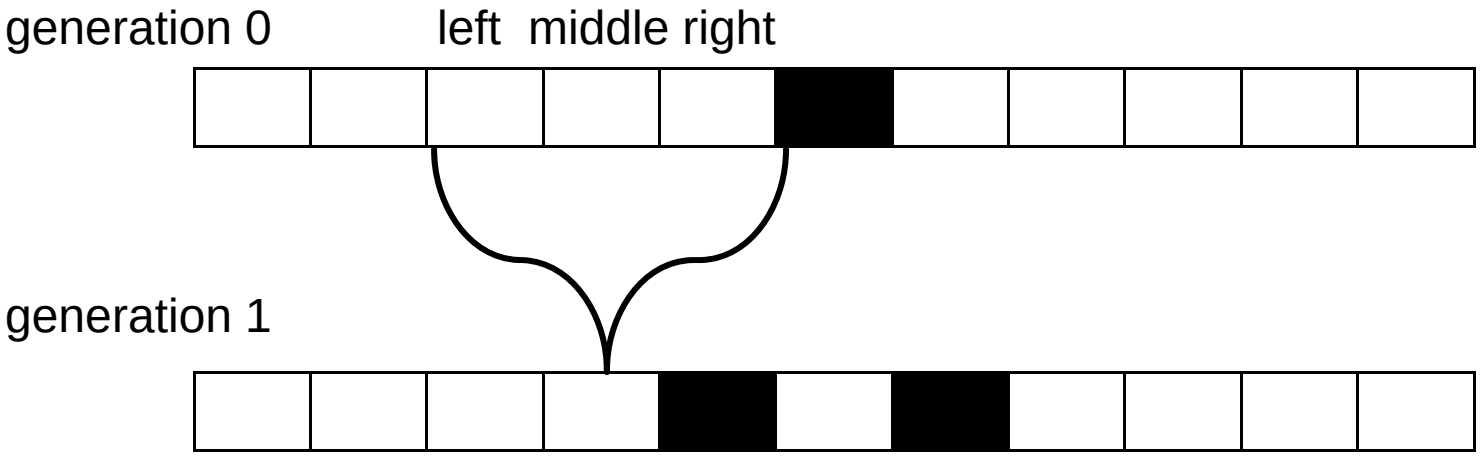
Example

- For each cell in the next generation, look at the 3 cells on the row immediately above it (immediately above, above-and-to-the-left, and above-and-to-the-right) in the previous generation.

- If the middle is white and either the left or the right is black (but not both), then this cell will become black in the next generation. Otherwise, it will be white.

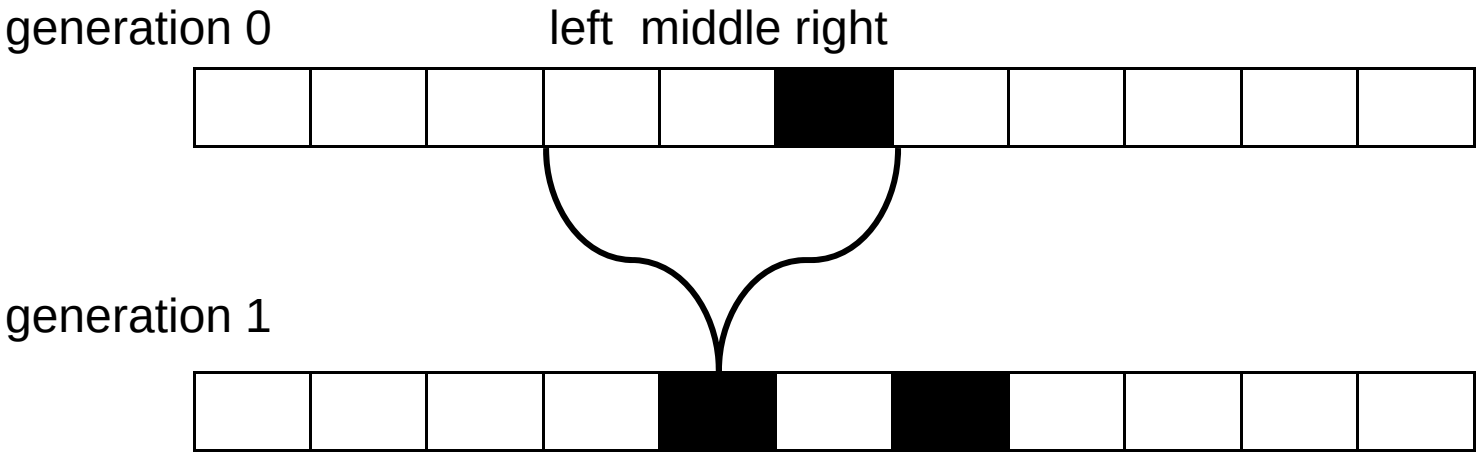
How it works

If the middle is white and either the left or the right is black (but not both) in the previous generation, then this cell will become black in the next generation. Otherwise, it will be white.



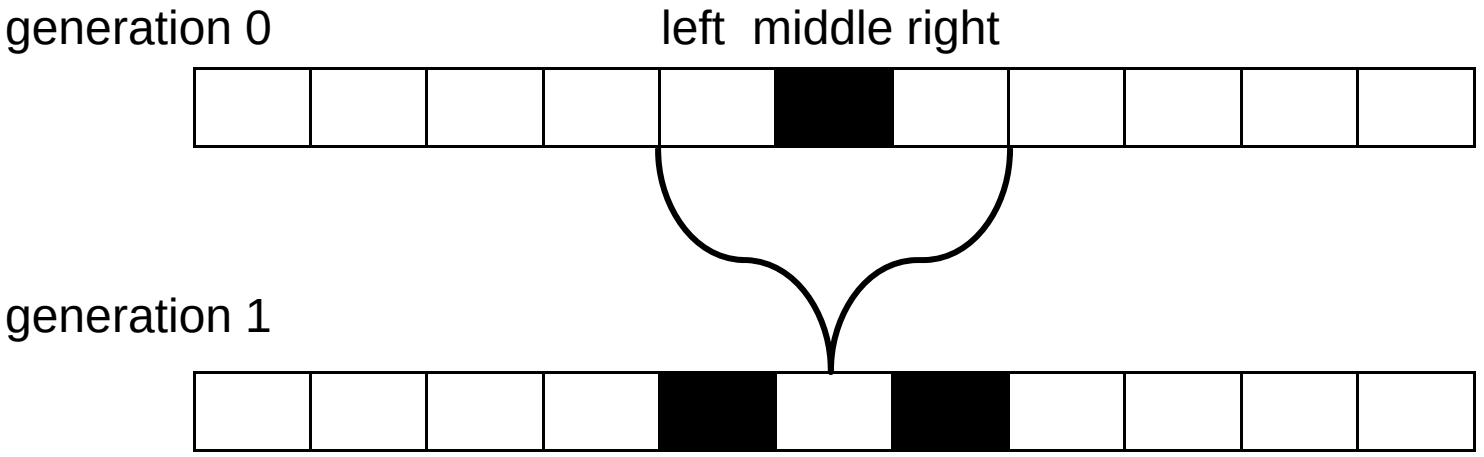
How it works

If the middle is white and either the left or the right is black (but not both) in the previous generation, then this cell will become black in the next generation. Otherwise, it will be white.



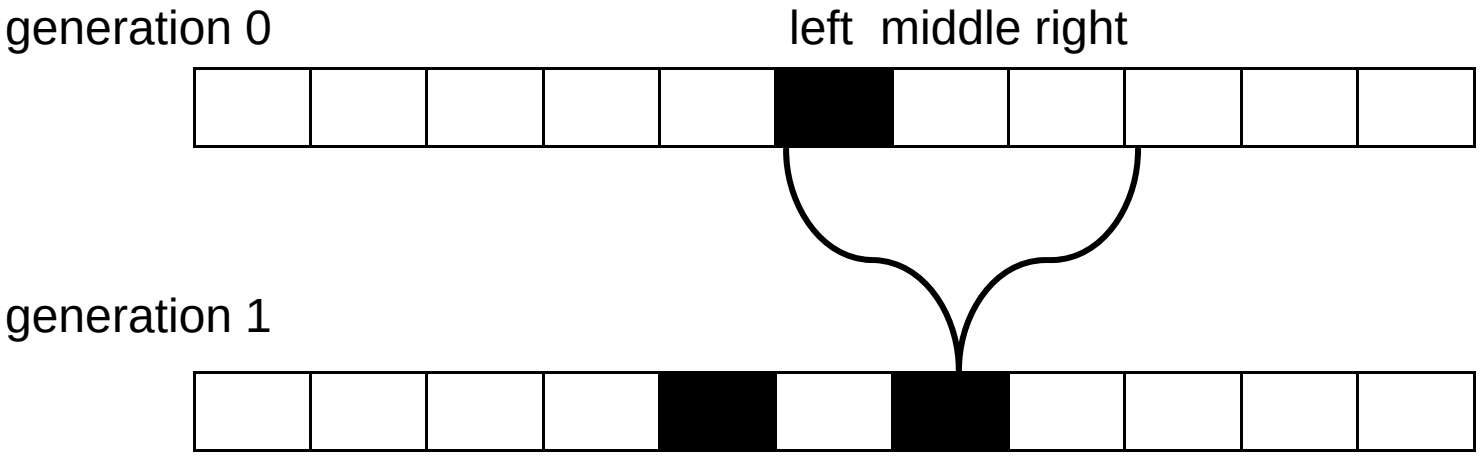
How it works

If the middle is white and either the left or the right is black (but not both) in the previous generation, then this cell will become black in the next generation. Otherwise, it will be white.



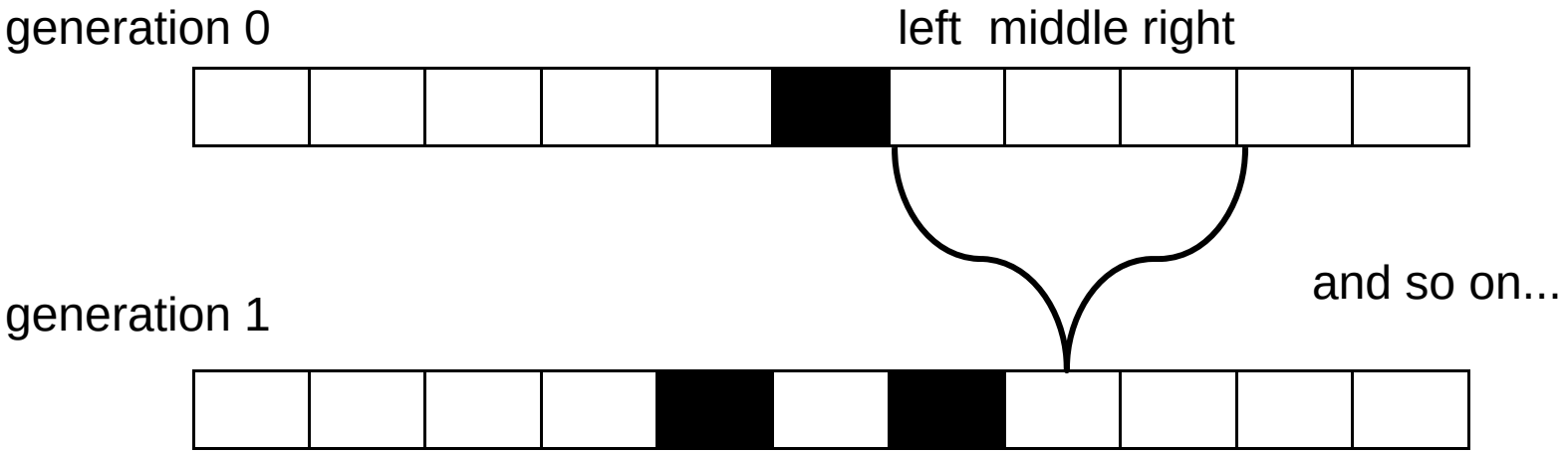
How it works

If the middle is white and either the left or the right is black (but not both) in the previous generation, then this cell will become black in the next generation. Otherwise, it will be white.



How it works

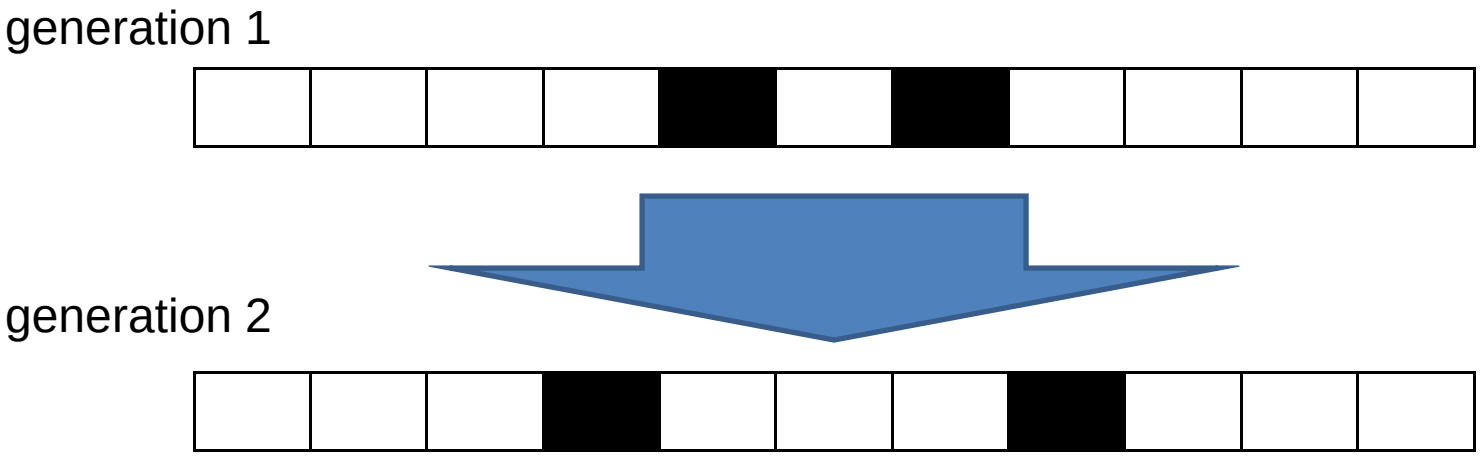
If the middle is white and either the left or the right is black (but not both) in the previous generation, then this cell will become black in the next generation. Otherwise, it will be white.



How it works

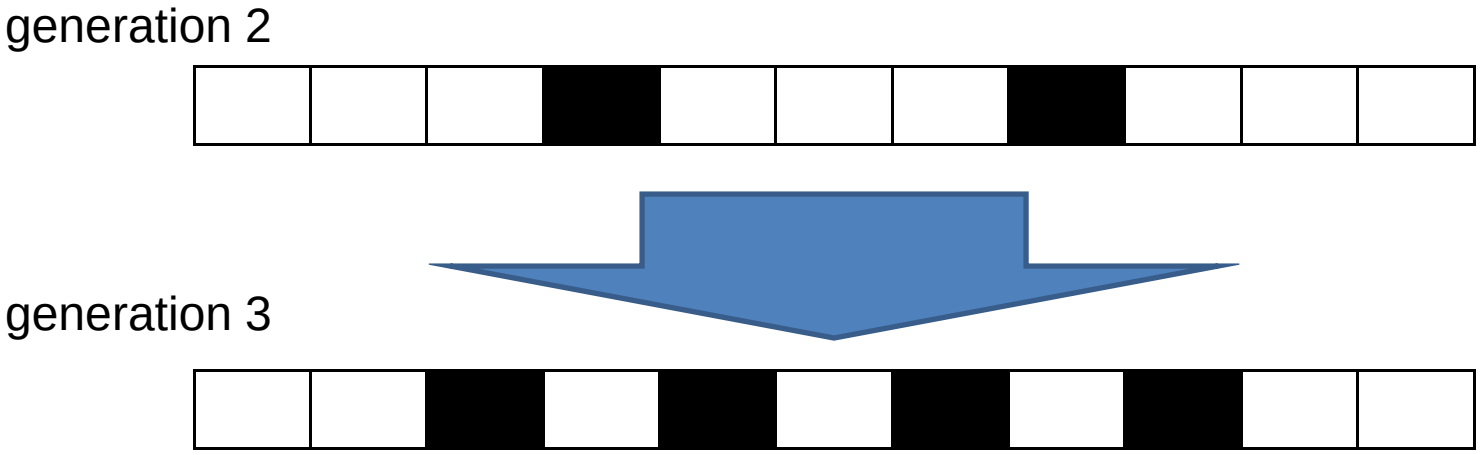
Once the next generation is created, use that to create a new generation.

If the middle is white and either the left or the right is black (but not both) in the previous generation, then this cell will become black in the next generation. Otherwise, it will be white.



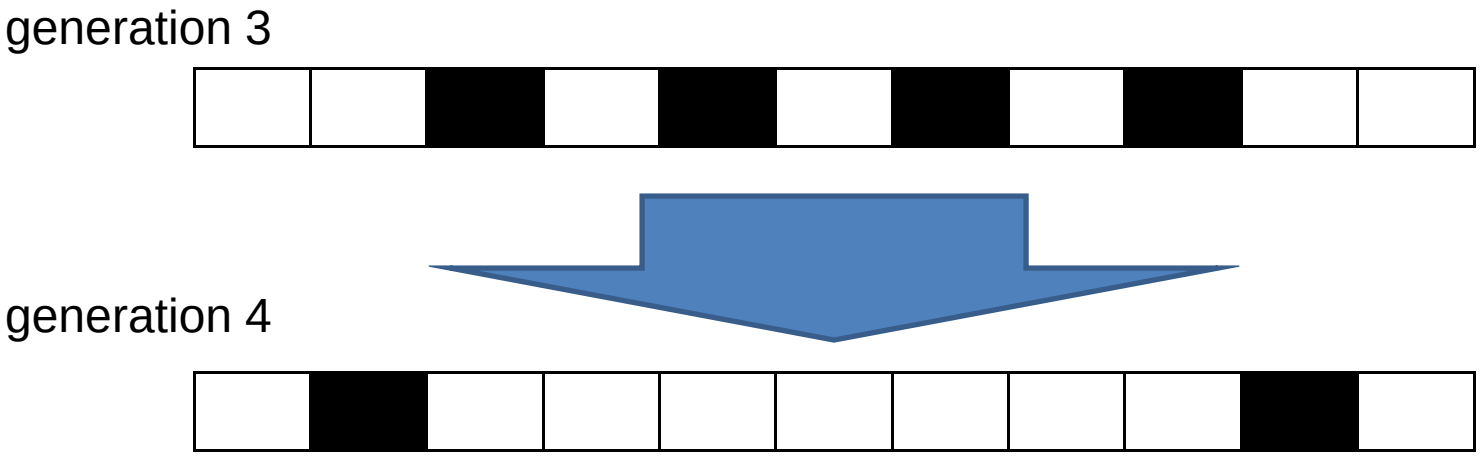
How it works

If the middle is white and either the left or the right is black (but not both) in the previous generation, then this cell will become black in the next generation. Otherwise, it will be white.



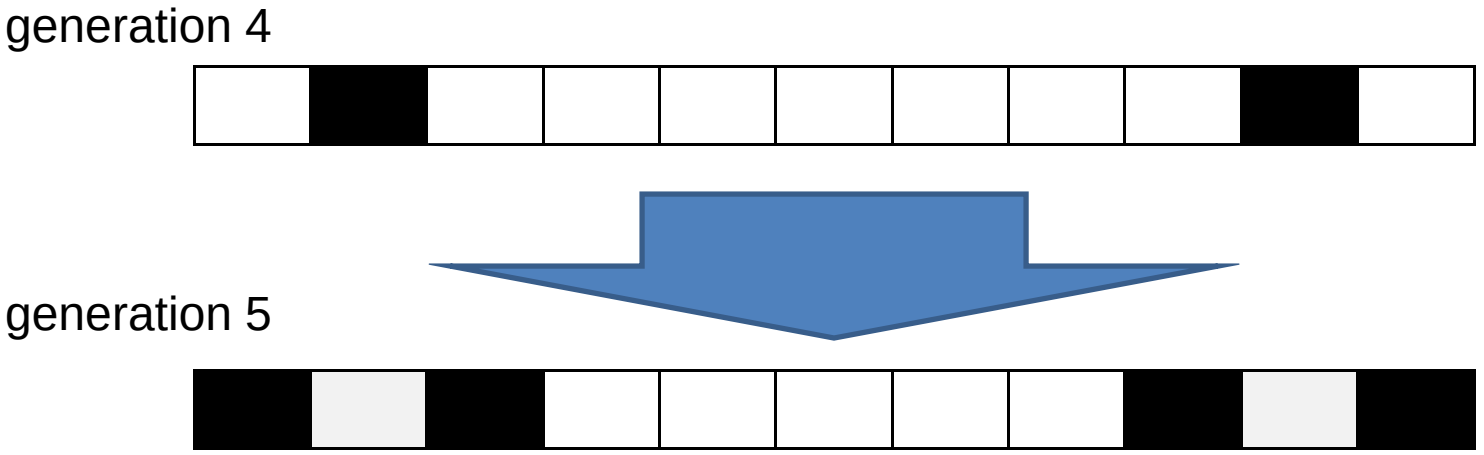
How it works

If the middle is white and either the left or the right is black (but not both) in the previous generation, then this cell will become black in the next generation. Otherwise, it will be white.

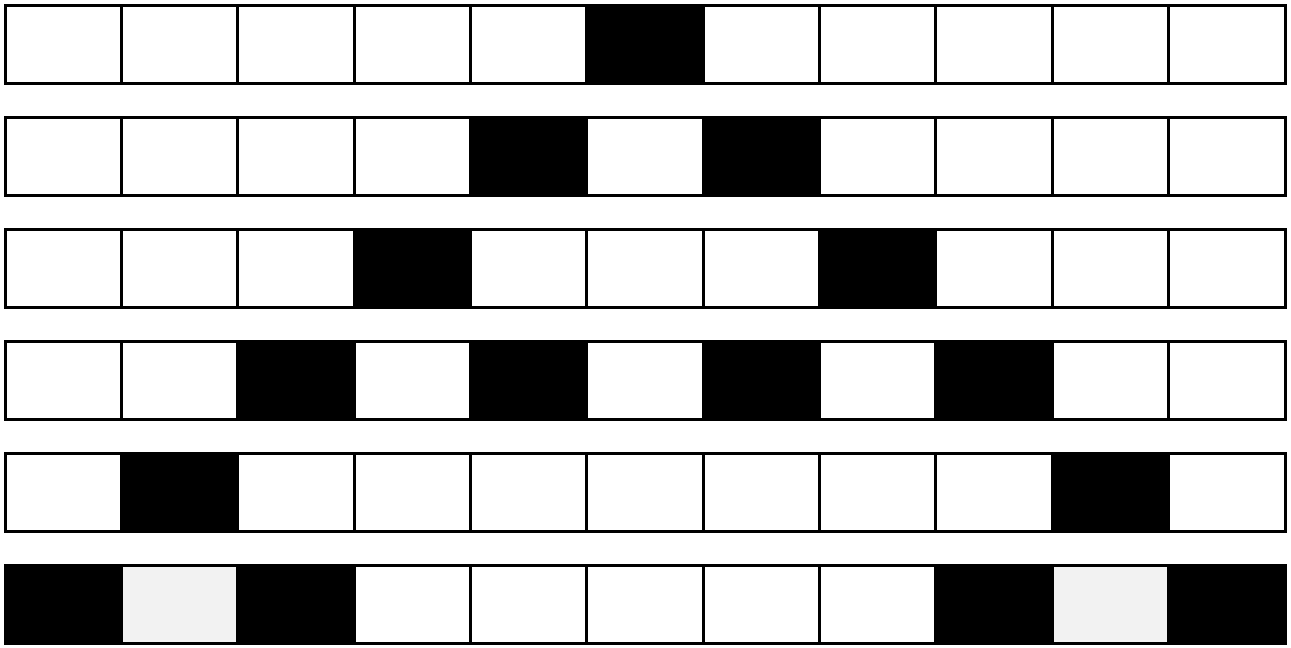


How it works

If the middle is white and either the left or the right is black (but not both) in the previous generation, then this cell will become black in the next generation. Otherwise, it will be white.



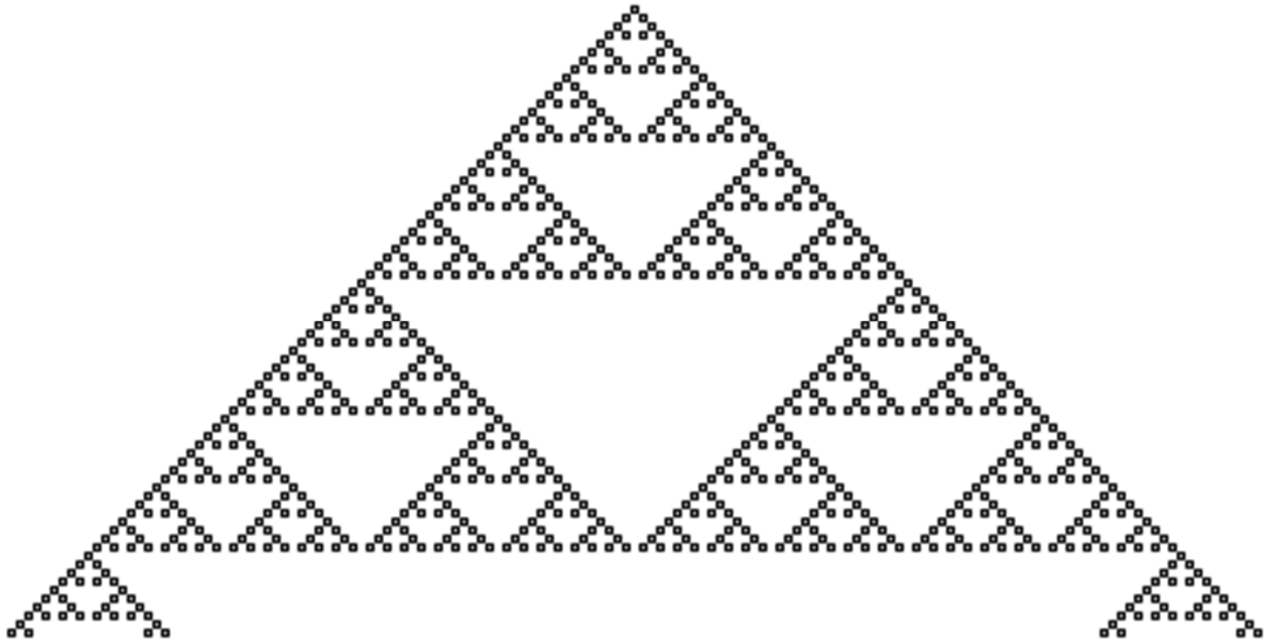
What we have so far



Keep going... what do we get?
(assume each row is infinite in length)

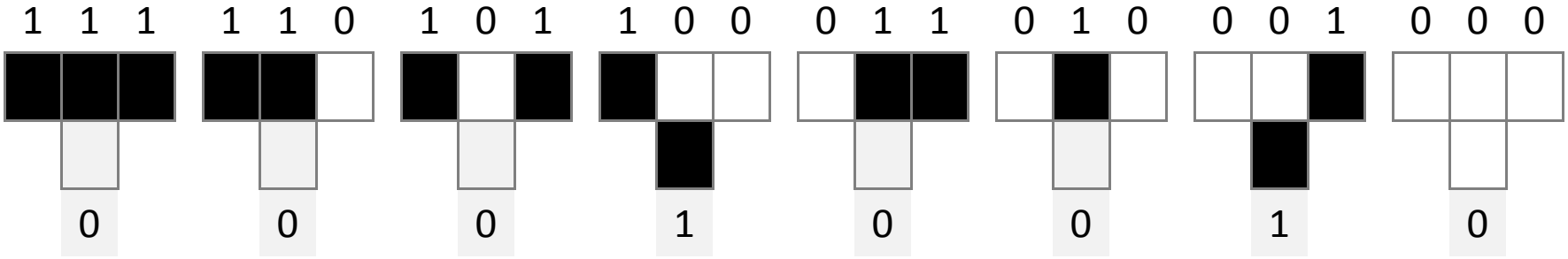
Results

Look familiar?



Rule 18

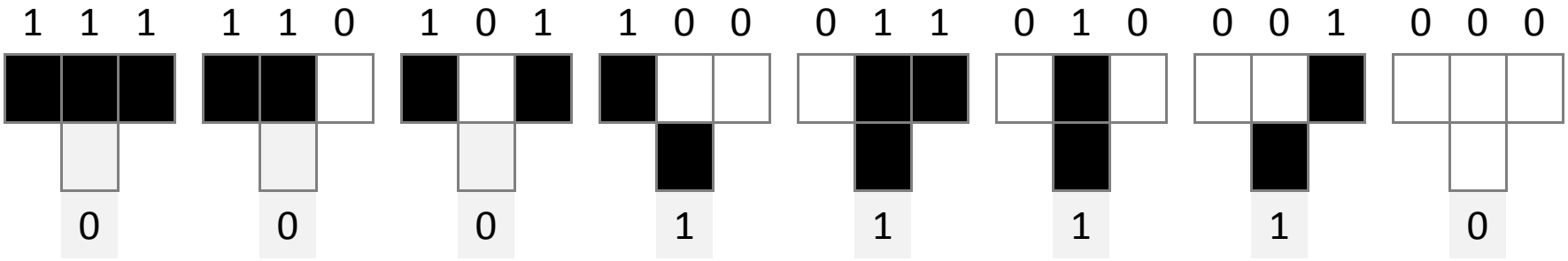
- This is known as “Rule 18” for 1-dimensional cellular automata.
  - Rule: If the middle is white and either the left or the right is black (but not both), then this cell will become black. Otherwise, it will be white.



00010010 in binary = 18

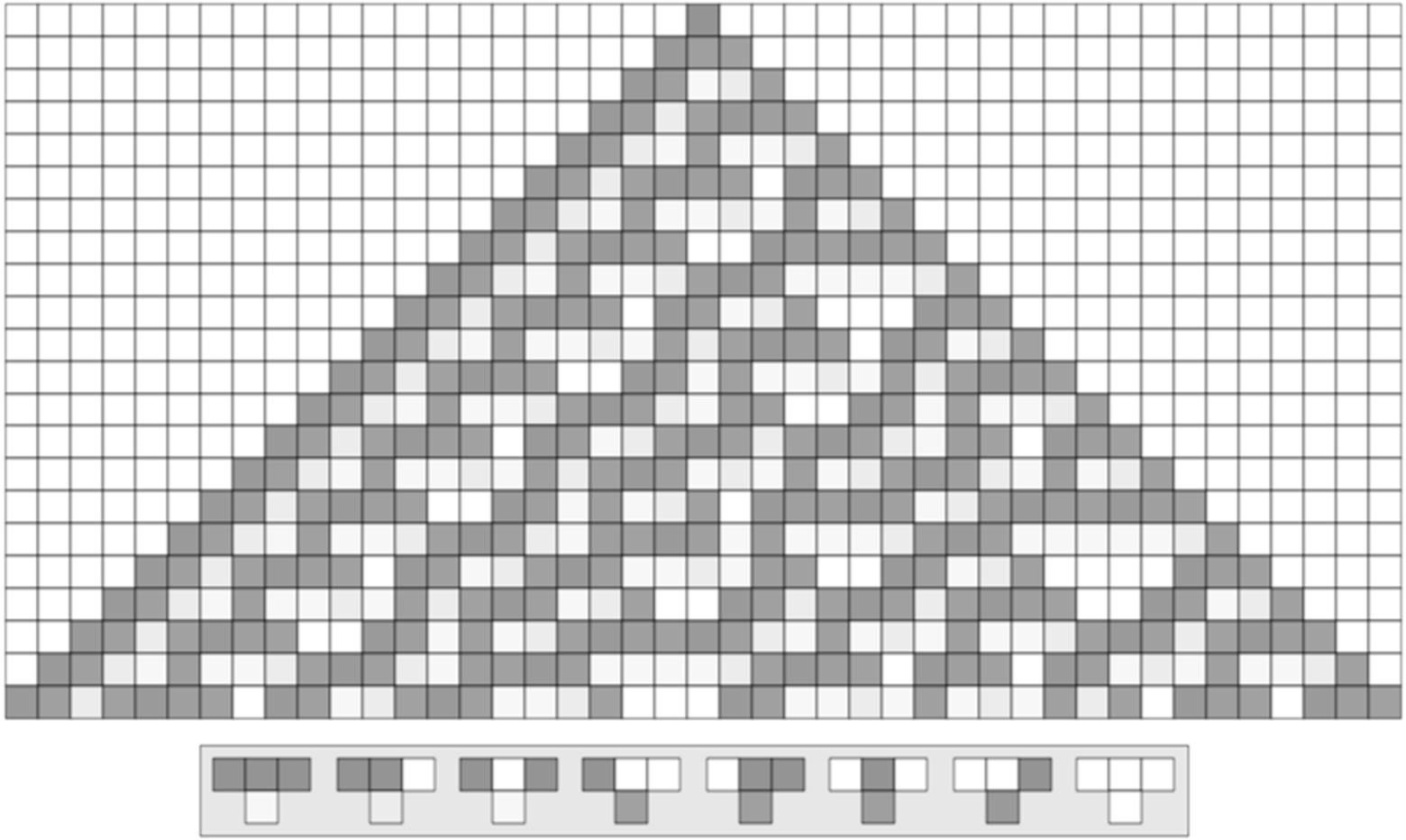
Rule 30

- How would you describe this rule?
- Try this rule using a random initial phase.
- Try this rule with a single black cell in the center.

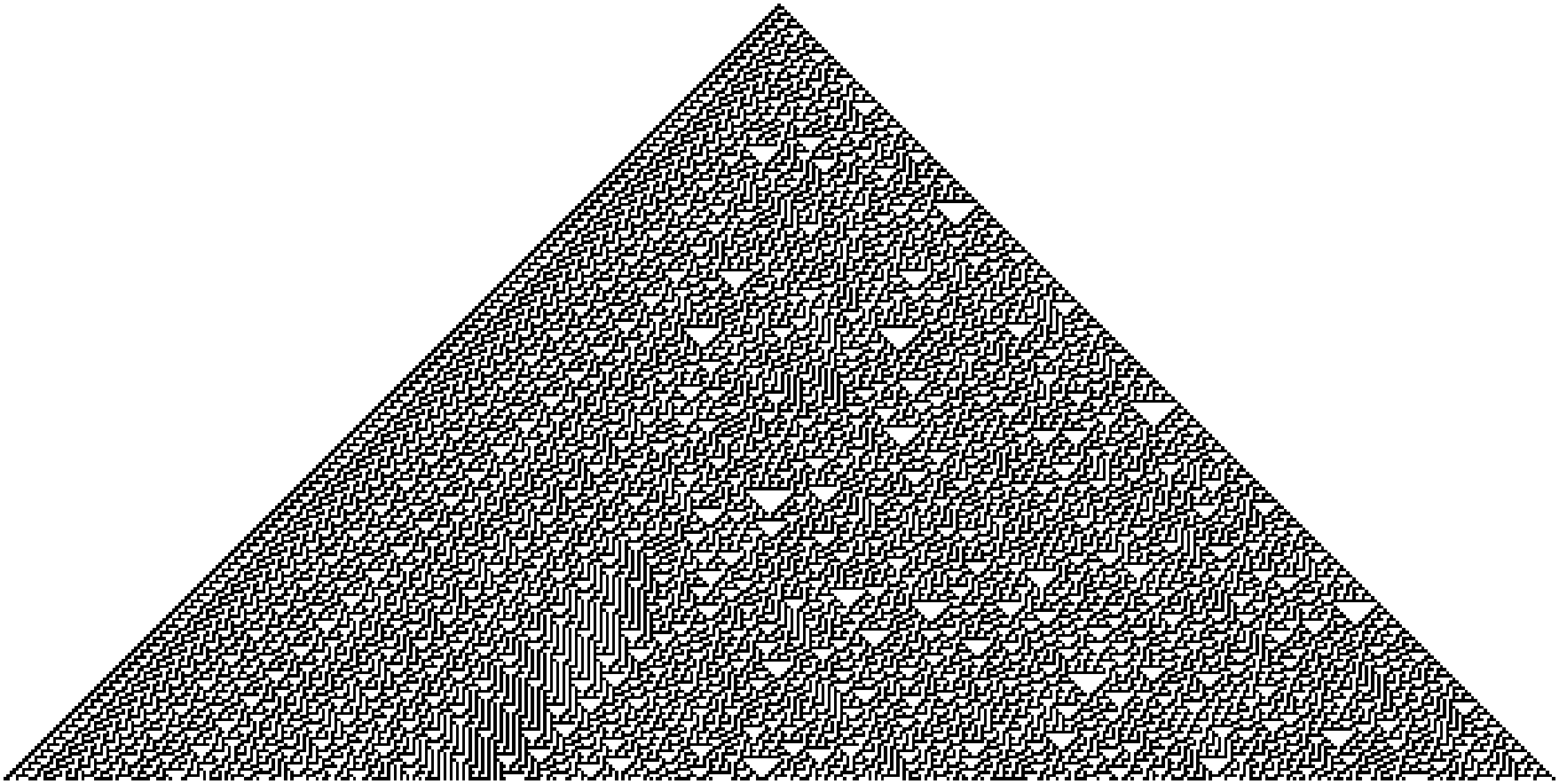


00011110 in binary = 30

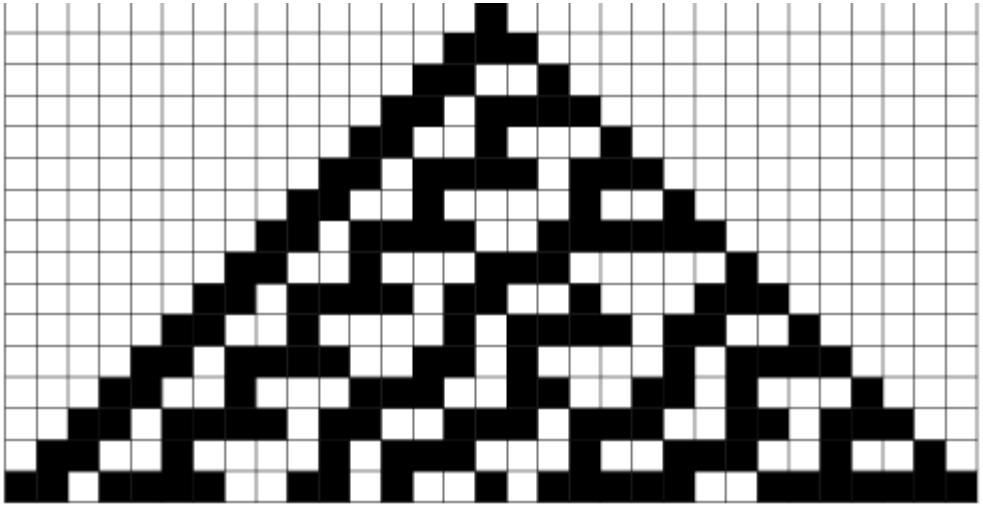
Rule 30



Rule 30



Rule 30



Rule 30 is of special interest : is used as the random number generator used for large integers in *Mathematica* (Wolfram 2002, p. 317).

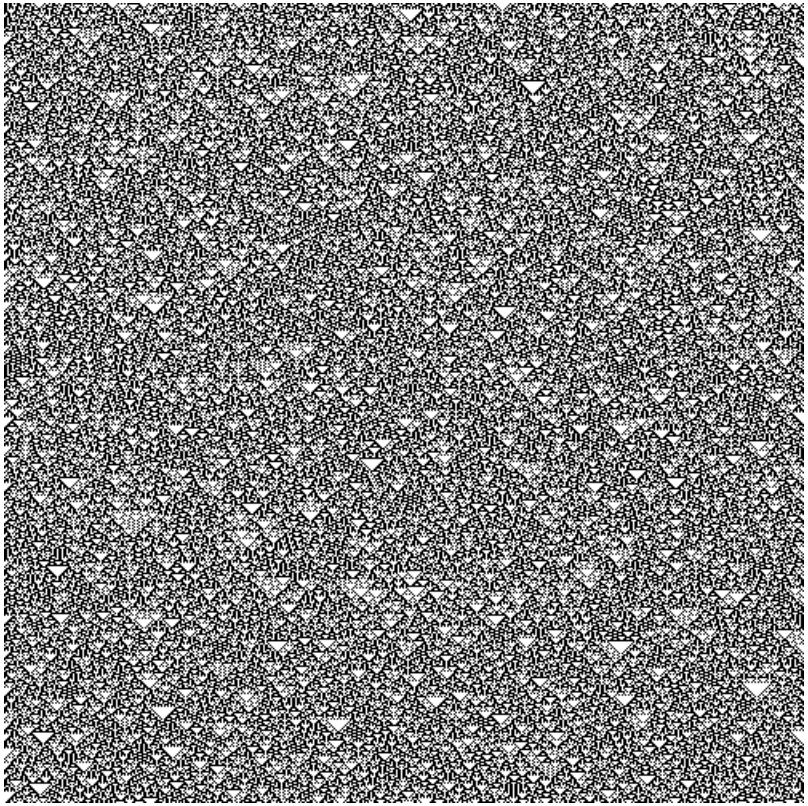
Central column given by 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, ...

Interpreting the central column as binary numbers and taking successive bits gives the sequence of numbers 1, 3, 6, 13, 27, 55, 110, 220, 441, 883, 1766, ...

Rule 30 in Nature

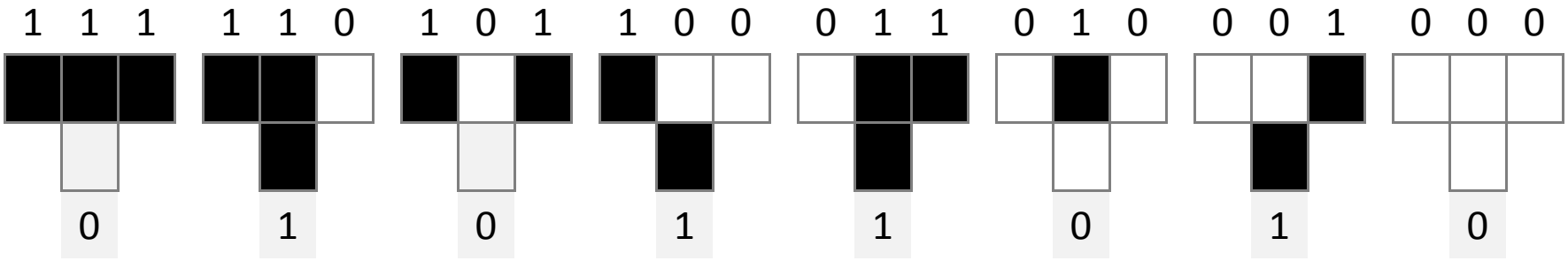
Rule 90

- Results starting with a random initial phase



Rule 90

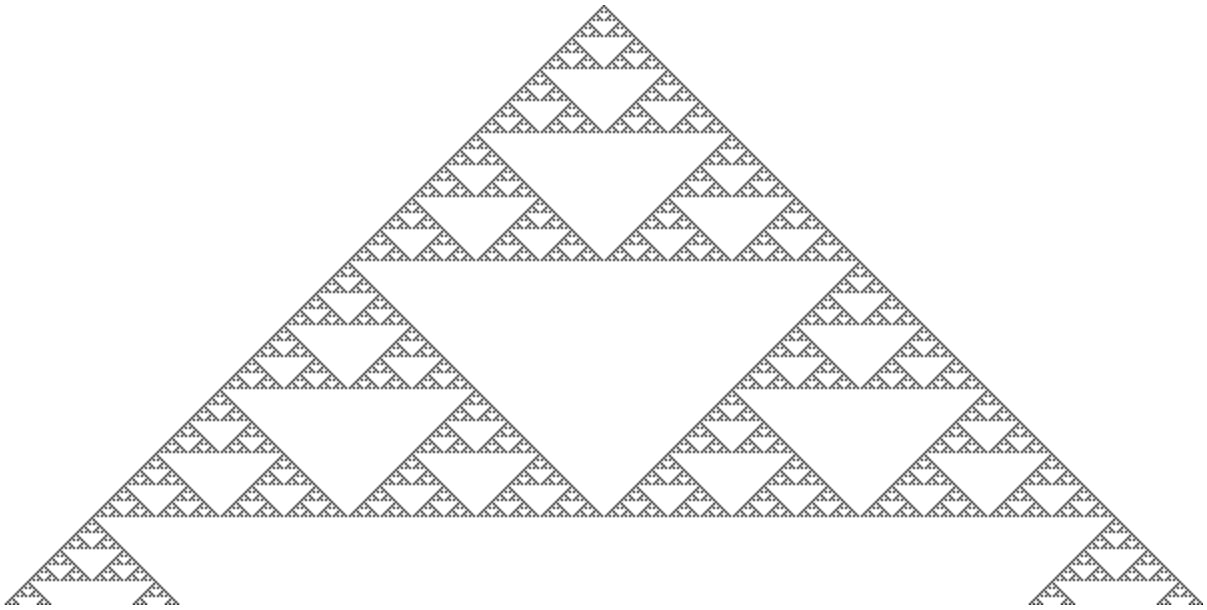
- How would you describe this rule?
- Try this rule using a random initial phase.
- Try this rule with a single black cell in the center.



01011010 in binary = 90

Rule 90

- Results starting with a single cell in the center of the first phase

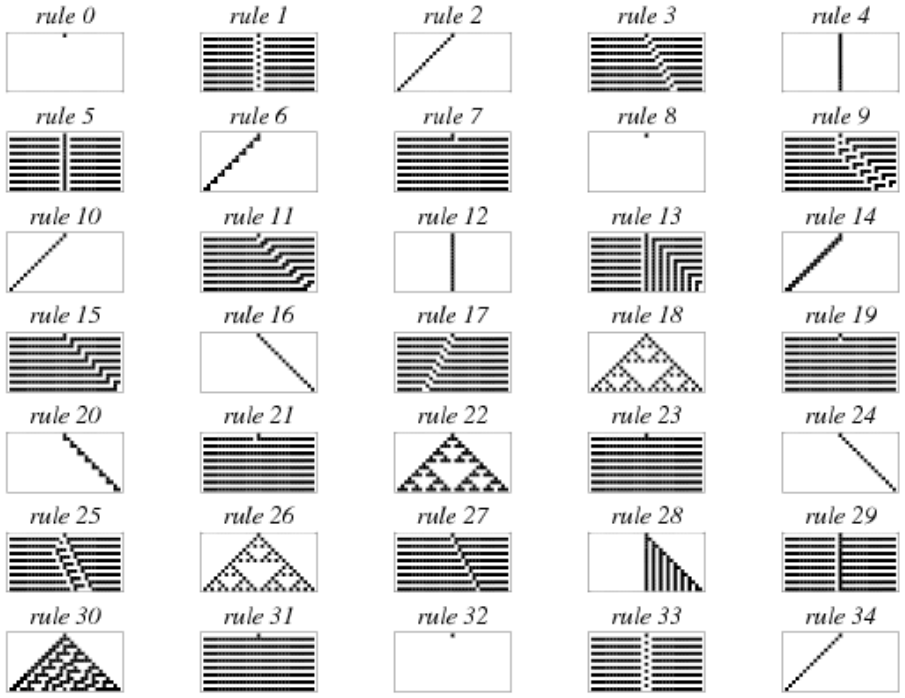


Rule Space

- How many rules are there for 1-dimensional automata? Recall that we have 8 bits to represent rules.
  - Answer: 256

Designing Rules

Assuming we start from a cell in the middle

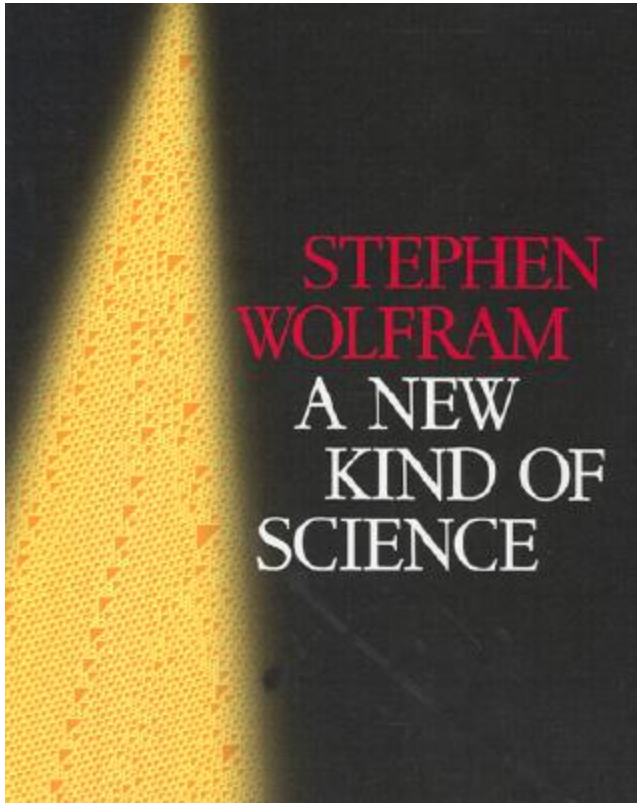


... and so on until Rule 255

Source: http://mathworld.wolfram.com/ElementaryCellularAutomaton.html

Cellular Automata

- For more information:



Game of Life

- An infinite two-dimensional cellular automaton devised by the mathematician John Horton Conway.
- The automaton consists of an infinite two-dimensional orthogonal grid of square cells, each of which is in one of two possible states, alive (■) or dead (□).
- Every cell interacts with its eight neighbors, which are the cells that are horizontally, vertically, or diagonally adjacent.

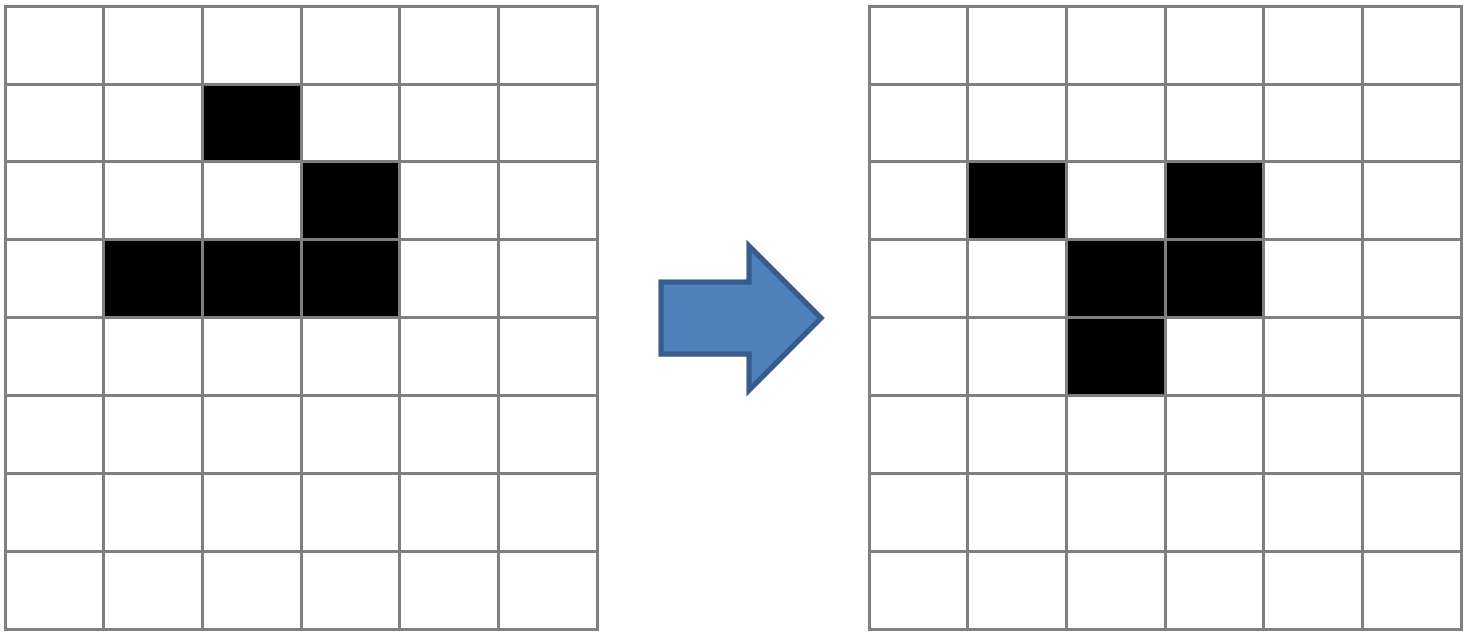
Game of Life: Rules

- At each step in time, the following transitions occur:
  - Any live cell with fewer than two live neighbors dies, as if caused by under-population.
  - Any live cell with two or three live neighbors lives on to the next generation.
  - Any live cell with more than three live neighbors dies, as if by overcrowding.
  - Any dead cell with exactly three live neighbors becomes a live cell, as if by reproduction.

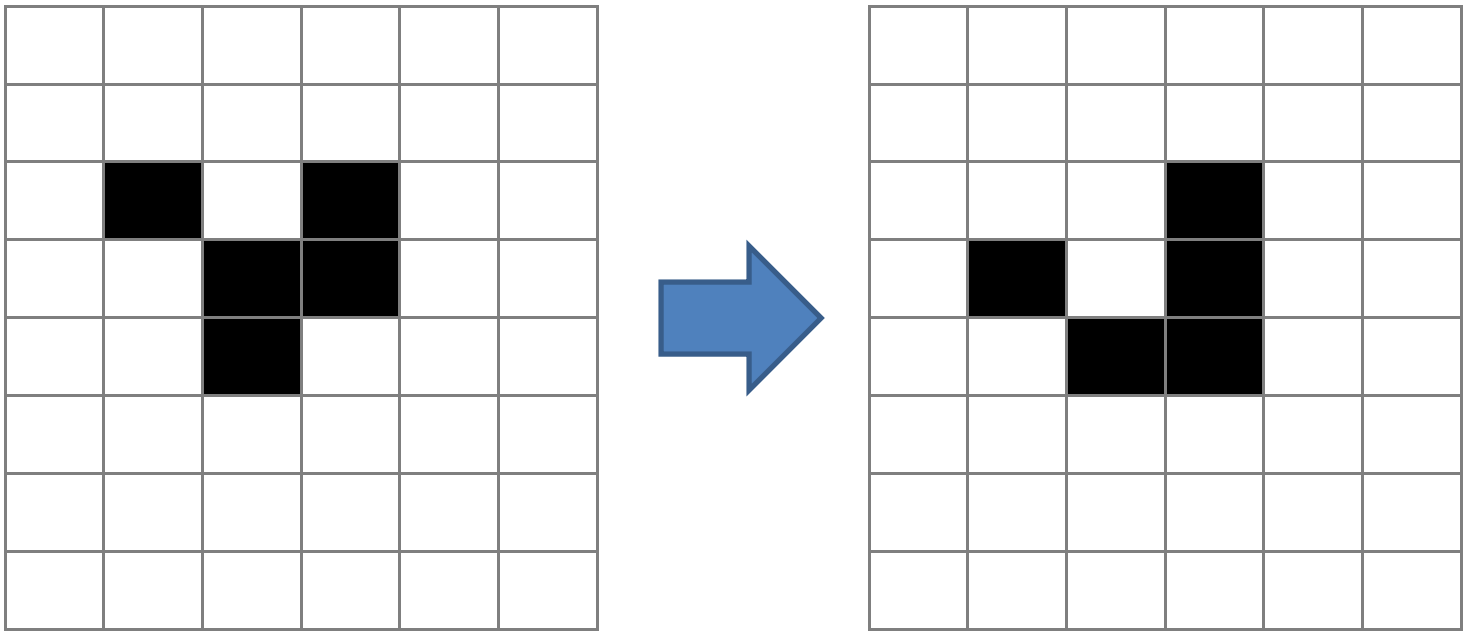
Generations

- The initial pattern constitutes the *seed* of the system.
- The first generation is created by applying the above rules simultaneously to every cell in the seed—births and deaths occur simultaneously, and the discrete moment at which this happens is sometimes called a *tick*.
- The rules continue to be applied repeatedly to create further generations.

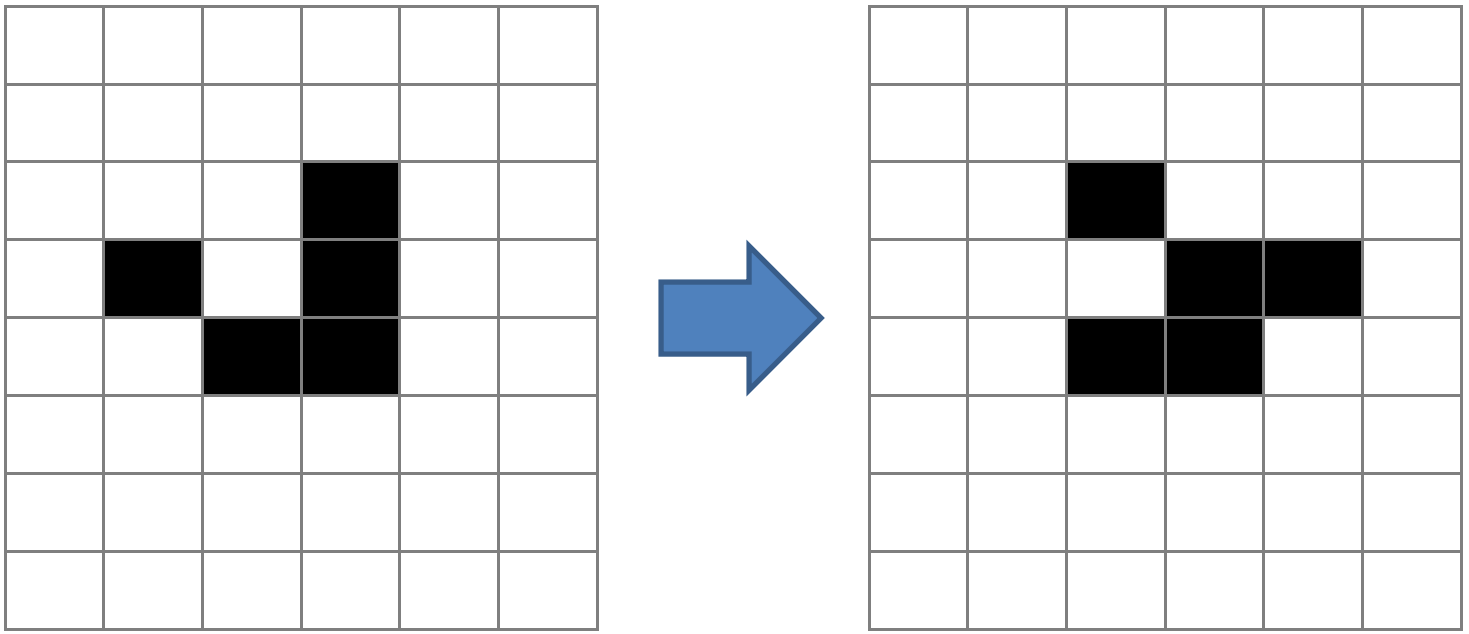
Example: Generation 1



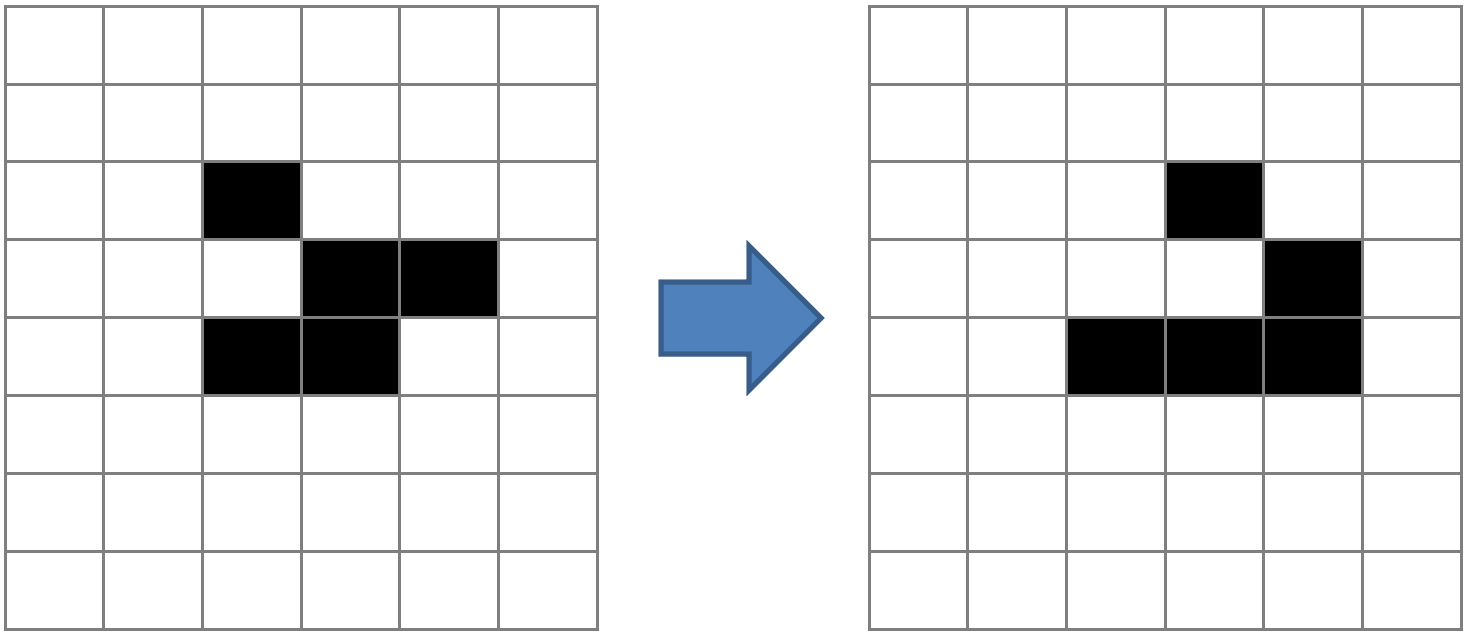
Example: Generation 2



Example: Generation 3



Example: Generation 4



Look familiar?

Game of Life and Randomness

- It was observed early on in the study of the Game of Life that random starting states all seem to stabilize eventually.
- Conway offered a prize for any example of patterns that grow forever. Conway's prize was collected soon after its announcement, when two different ways were discovered for designing a pattern that grows forever.
(from www.math.com)

Universality

- Anything that can be computed algorithmically with a computer with unlimited memory and no time constraints can be computed within Conway's Game of Life

Simulating Cellular Automata

- You can experiment with the CA simulator at

  http://www.cs.cmu.edu/~15110s13/resources.html