COMP112 in Week 8, 2022

Cellular Automata

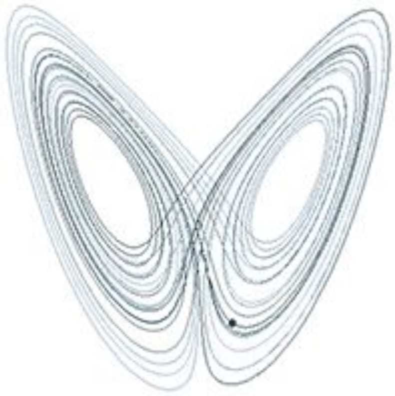
simple local computation is surprisingly powerful

Chaos and Order

Examples of chaos like this seem to require continuous "space"

But we will see that essentially the same phenomena can occur in incredibly simple **automata**, with "discrete" states

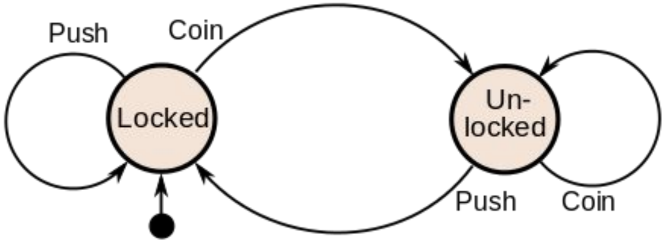
These automata exhibit very rich behaviour indeed



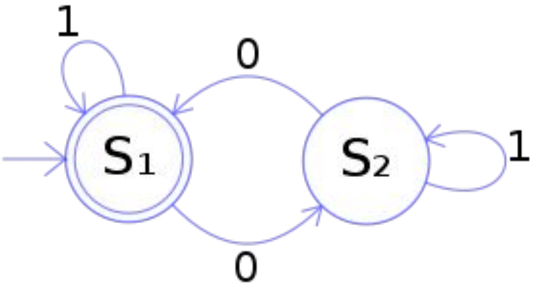
Chaos, in the Lorenz system – an example of the *Butterfly effect*

We've met Automata

This finite state machine represents a locker



This one acts as an "acceptor" for binary numbers that contain an even number of zeros

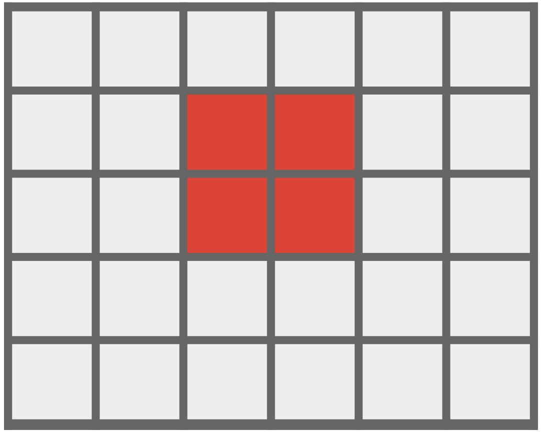


Cellular Automata

Conway's Game of Life

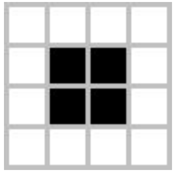
1) an alive cell dies if it has
   < 2 live neighbours, or
   > 3 live neighbours,

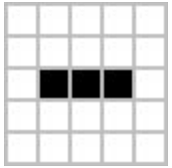
2) a dead cell with exactly 3 live
   neighbours comes to life

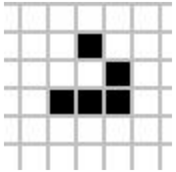


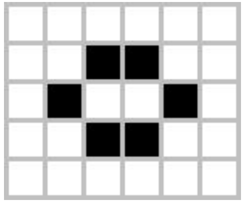
Amazingly enough, this automaton has the power of a universal Turing Machine: anything that can be computed by an algorithm can be computed within the "Game of Life".

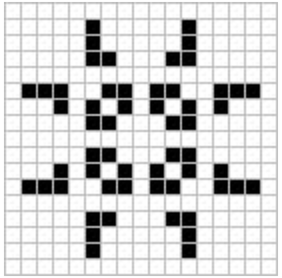
Game of Life

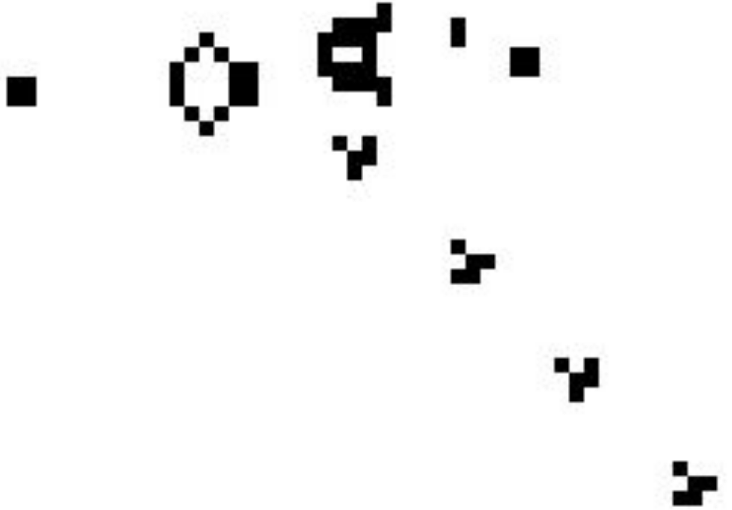












Nice: https://www.youtube.com/watch?v=Kk2MH9O4pXY

Game of Life

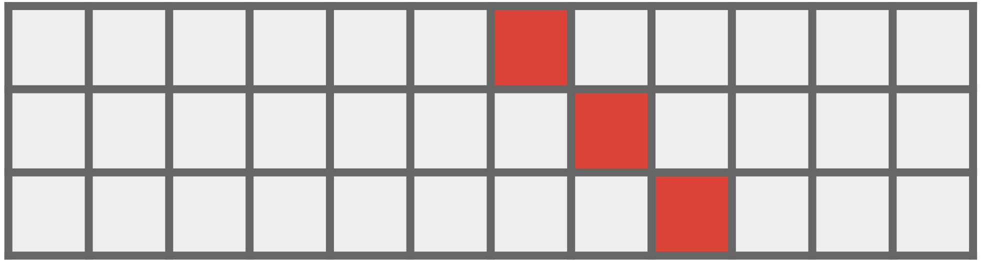
The Game of Life is **undecidable**: given an initial pattern and a later pattern, no algorithm exists that can tell whether the later pattern is ever going to appear.
This is a corollary of the **halting problem**: the problem of determining whether a given program will finish running or continue to run forever from an initial input.

And that's just the start - see for example https://www.youtube.com/watch?v=C2vgICfQawE, or https://youtu.be/HeQX2HjkcNo (*Life* is 29:30 in, but the whole thing is fantastic)

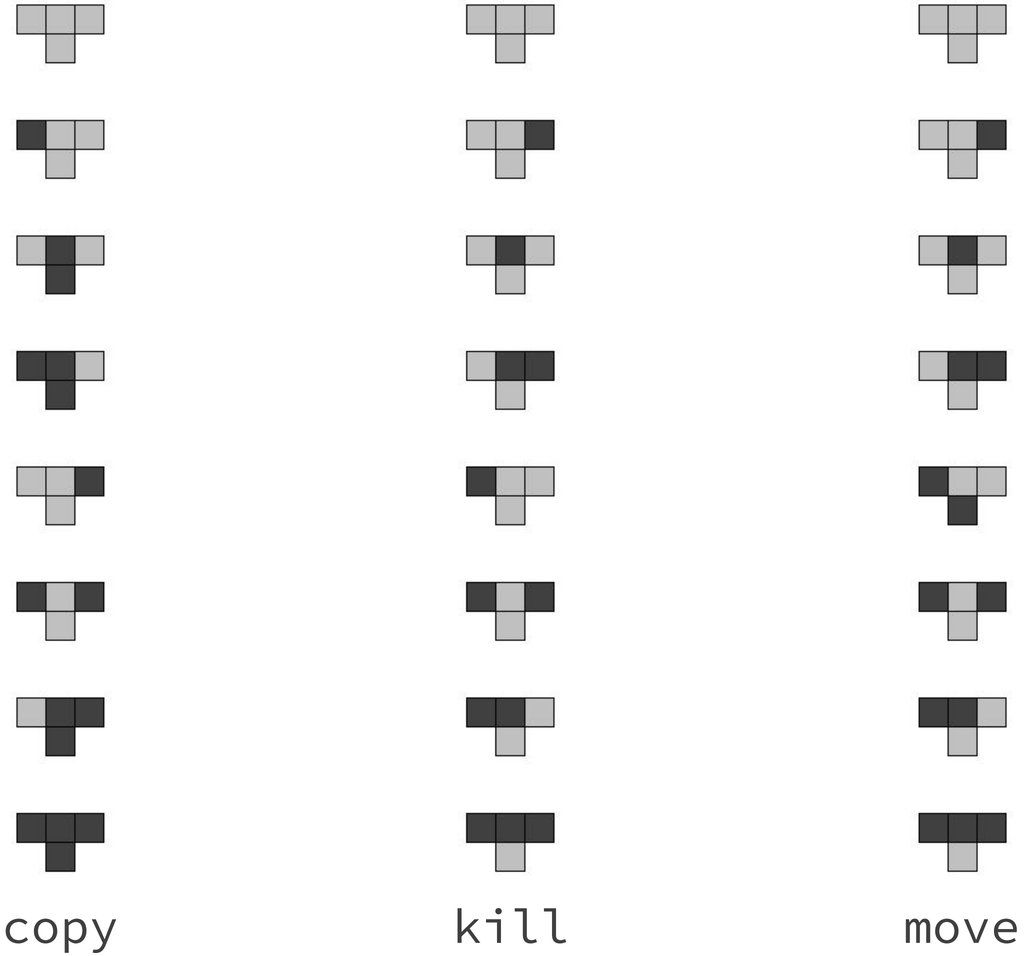
How simple can you go, and still have emergent complexity?

Let’s try just ONE dimension

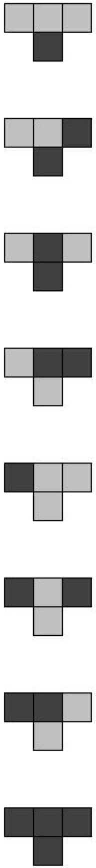
This has the advantage that we can “unroll time” down the page



Some rules



Another Rule



0: 000 → 1

1: 001 → 1

2: 010 → 1

3: 011 → 0

4: 100 → 0

5: 101 → 0

6: 110 → 0

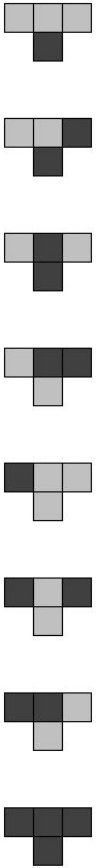
7: 111 → 1

In binary, this could be called “rule 10000111”

TOTAL: 135

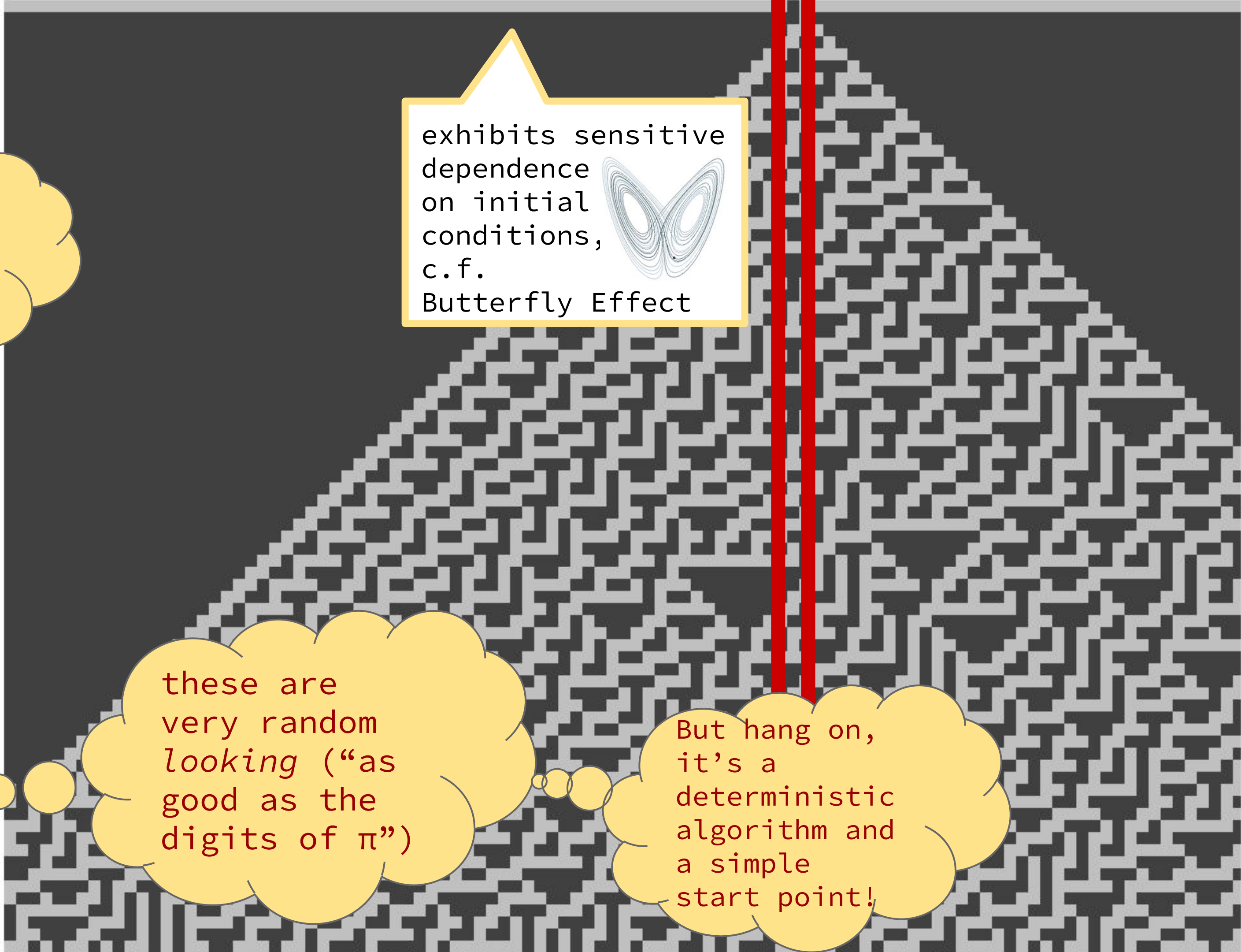
Wolfram Alpha “Rule 135”

Rule 135



same as
Rule 30
(black-white
reversed)

1
1
0
1
1
1
0
0
1
1
0
0
0
1
0
1
1
0
0
1
0
0
1
1
1
0
1
1
0
1
1
…



exhibits sensitive dependence on initial conditions, c.f. Butterfly Effect

these are very random *looking* ("as good as the digits of π")

But hang on, it's a deterministic algorithm and a simple start point!

4 classes of CA

The actual sequence depends on the initial condition (i.e. 1^{st} line)

But each CA rule seems to fall into one of these classes:

I. the pattern stabilizes

II. mostly stable or oscillating structures

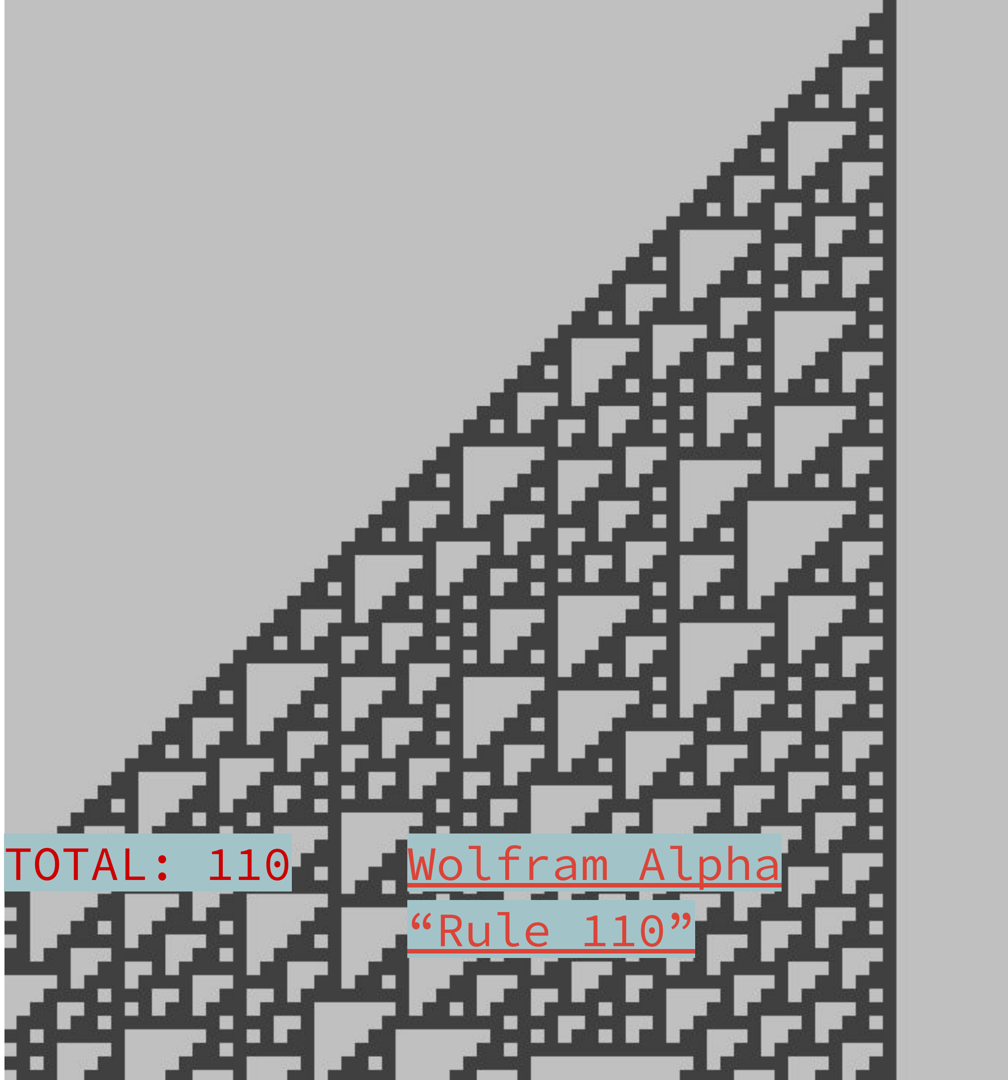
III. patterns evolve in a seemingly chaotic fashion

IV. automata in which patterns become extremely complex and may last for a long time, with stable local structures.

This last class is capable of simulating a Turing Machine.

A "Turing Complete" Rule

- 0: 000 → 0
- 1: 001 → 1 * 2^1=2
- 2: 010 → 1 * 2^2=4
- 3: 011 → 1 * 2^3=8
- 4: 100 → 0
- 5: 101 → 1 * 2^5=32
- 6: 110 → 1 * 2^6=64
- 7: 111 → 0



So What?

https://en.wikipedia.org/wiki/A_New_Kind_of_Science

"Simply enumerating all possible variations of almost any class of programs quickly leads one to examples that do unexpected and interesting things. This leads to the question: if the program is so simple, where does the complexity come from?

In a sense, there is not enough room in the program's definition to directly encode all the things the program can do. Therefore, simple programs can be seen as a minimal example of emergence.

A logical deduction from this phenomenon is that if the details of the program's rules have little direct relationship to its behavior, then it is very difficult to directly **engineer** a simple program to perform a specific behavior."

TOMORROW'S MISSION

Explore one-dimensional cellular automata, by building a an interactive program.

We will see:

- how to use the mouse to select points on the canvas
- a 2-dimensional array