Complex Systems and Network Science:

Models

Ozalp Babaoglu
Dipartimento di Informatica — Scienza e Ingegneria
Università di Bologna
www.cs.unibo.it/babaoglu/

Why model?

- Models are abstractions of reality that serve two purposes:
  - *Explain* observed (past) behaviors
  - *Predict* unobserved (future) or unobservable behaviors
- Models help us
  - understand the world we live in
  - understand and use data by turning it into knowledge
  - make better decisions and designs
  - become better citizens (models are everywhere)
- What can be modeled?
  - Just about anything

Why model?

- In our daily lives, we rely on sophisticated ***mental models*** to perform many tasks: walk, ride a bicycle, drive a car, avoid collisions, hit a tennis ball, etc.
- These models are able to incorporate not only the ***physical world*** (Newtonian mechanics) but also ***economic***, ***social***, ***cultural clues***

Why model?

Why model?

Why model?

- To be useful, a model has to be *compact* and *simple* while maintaining *fidelity* to what is being modeled
- Abstract away the unnecessary details yet maintain the essence
- “Everything should be made as simple as possible, but no simpler” — *Albert Einstein* (1933)
- “Essentially, all models are wrong, but some are useful” — *George Box* (1987)

Tension between compactness and fidelity

Sylvie and Bruno Concluded: The Man in the Moon, by Lewis Carroll, 1889

“That’s another thing we’ve learned from your Nation,” said Mein Herr, “map-making. But we’ve carried it much further than you. What do you consider the largest map that would be really useful?”

“About six inches to the mile.”

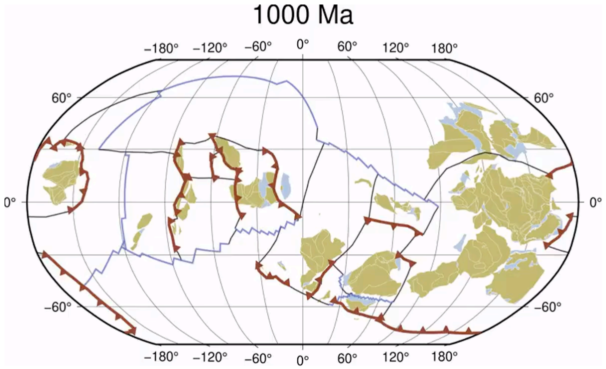
“Only six inches!” exclaimed Mein Herr. “We very soon got to six yards to the mile. Then we tried a hundred yards to the mile. And then came the grandest idea of all! We actually made a map of the country, on the scale of a mile to the mile!”

“Have you used it much?” I enquired.

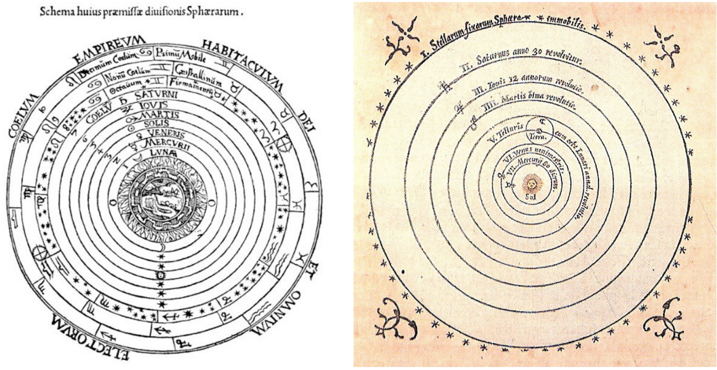
“It has never been spread out, yet,” said Mein Herr: “the farmers objected: they said it would cover the whole country, and shut out the sunlight! So we now use the country itself, as its own map, and I assure you it does nearly as well.

Well-known models: Geographic maps

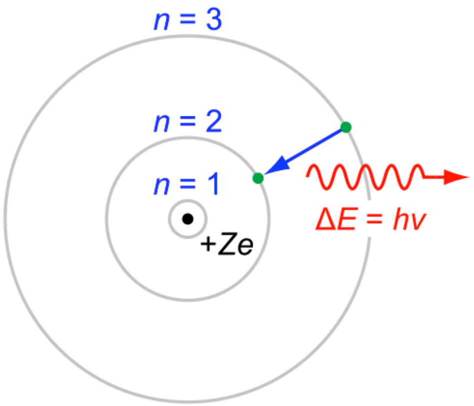
Well-known models:
Plate Tectonics



Well-known models:
Solar system

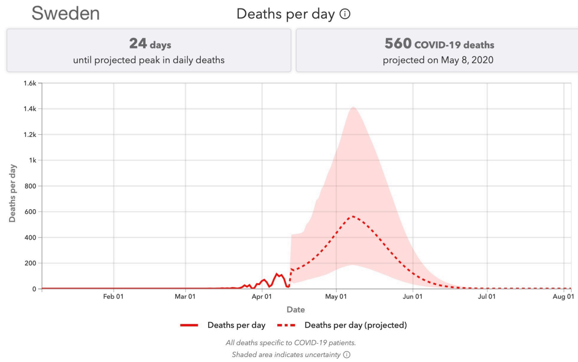


Well-known models:
Bohr's atom

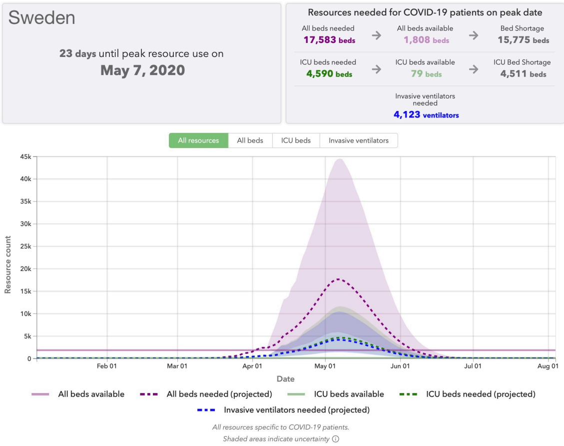


Well-known models:
Pandemics

- Modeling by the *Institute for Health Metrics and Evaluation* (IHME) at the University of Washington (as of 13 April 2020)

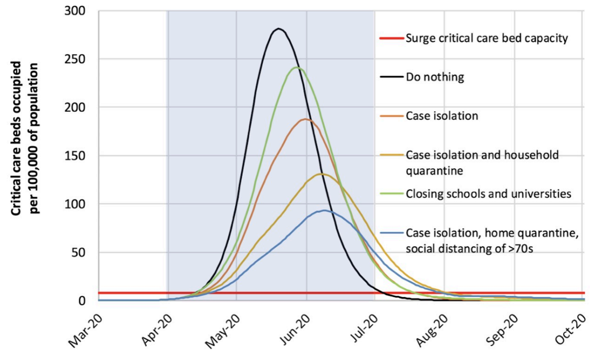


Well-known models: Pandemics



Well-known models: Pandemics

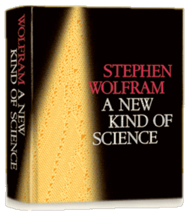
- Covid-19 peaks in ICU beds for different social distancing measures



Source: Impact of non-pharmaceutical interventions (NPIs) to reduce COVID19 mortality and healthcare demand, Neil Ferguson et. al, Imperial College

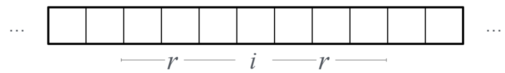
Models for Complex Systems Cellular automata

- Abstract model for simple individual behaviors and simple interactions leading to complex aggregate behaviors
- Developed by John von Neumann as a formal tool to study "mechanical self replication"
- Studied extensively by Stephen Wolfram



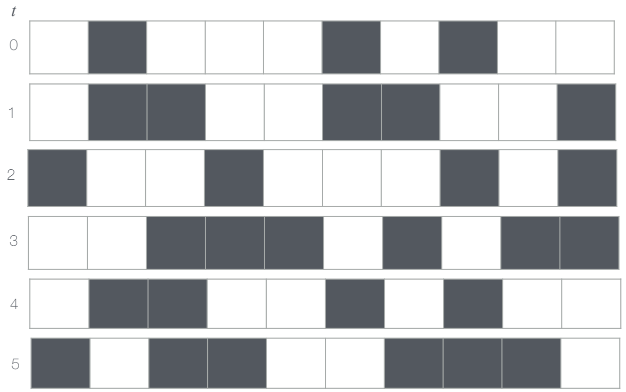
1-Dimensional CA

- An (infinite) array of "cells"
- Each cell has a value from a k-ary state
- Each cell has has a position i in the array and has r left and r right neighbors

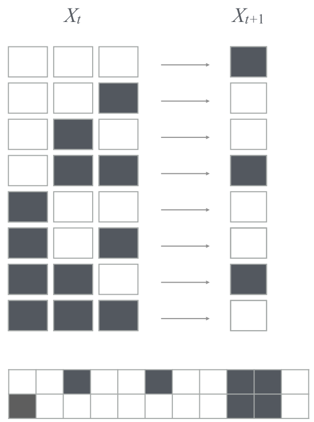


- State of a cell at time $t+1$ is a function of cell's state and its neighbors state at time t
- Assume $k = 2$ (binary state)
- Assume $r = 1$ (neighborhood of size 2)

1-Dimensional CA



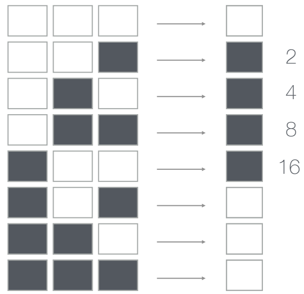
State transitions (Look-up table for r=1)



Wolfram canonical enumeration

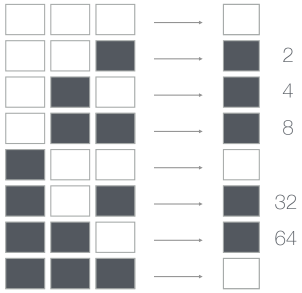
- With a binary state and radius $r = 1$, there are 2^{2^3}=256 possible CAs
- Read off the final state column of the look-up table as a binary number
- Each possible CA identified through an integer 0—255

Wolfram canonical enumeration



Rule = 30

Wolfram canonical enumeration



Rule = 110

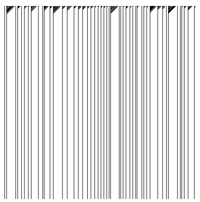
Wolfram's classification

- **Class I**: Nearly all initial patterns evolve quickly into a stable, homogeneous state (fixed point)
- **Class II**: Nearly all initial patterns evolve quickly into stable or oscillating structures (periodic)
- **Class III**: Nearly all initial patterns evolve in a pseudo-random or chaotic manner (chaotic)
- **Class IV**: Nearly all initial patterns evolve into structures that interact in complex and interesting ways (complex — capable of universal computation)

Wolfram's classification: Class I

Rule 40

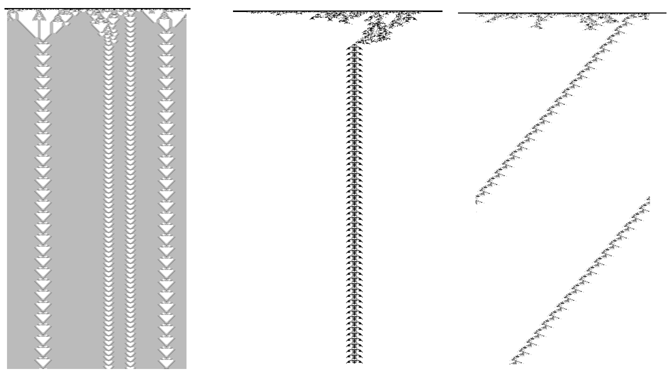
Rule 172



Rule 234

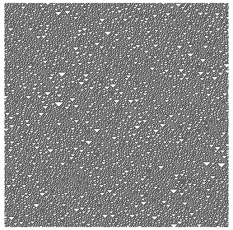
Source: https://plato.stanford.edu/entries/cellular-automata/supplement.html

Wolfram's classification: Class II

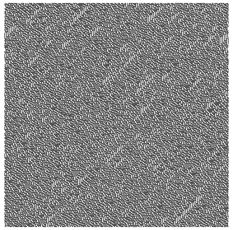


Wolfram's classification: Class III

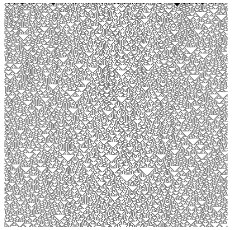
Rule 30



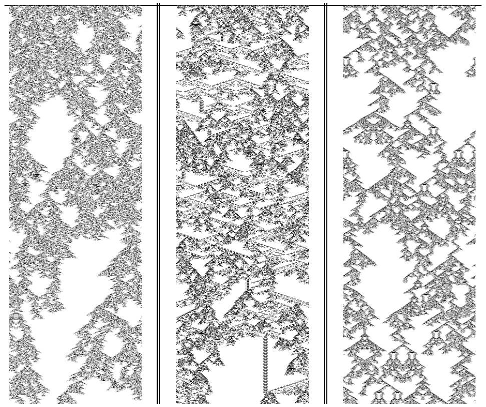
Rule 101



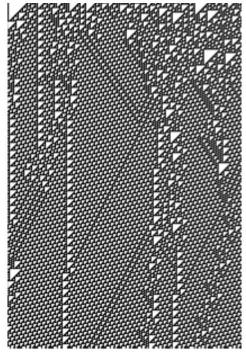
Rule 146



Wolfram's classification: Class IV



Rule 110



NetLogo

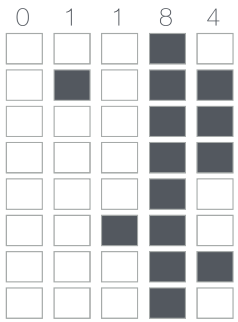
- Library/Computer Science/CA 1D Elementary
- ElementaryCAs

CAs as dynamical systems

- CAs are discrete-time, deterministic dynamical systems that exhibit fixed-point, periodic and chaotic behavior
- Similar to the Logistic Map except that the state variable for CAs is discrete (while continuous for the Logistic Map)
- For the Logistic Map, there is a control parameter R
- What is the equivalent for CAs?

Langdon's λ metric

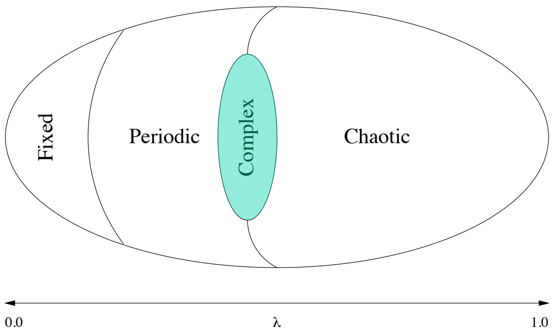
- Seek a compact characterization of the CA behavior class
- Count the number of "ones" in the look-up table final state column



Langdon's λ metric

λ	All Rules	Class III	Class IV	Normalized λ
0	1	0	0	0
1	8	0	0	0,125
2	28	2	0	0,25
3	56	4	1	0,375
4	70	20	4	0,5
5	56	4	1	0,625
6	28	2	0	0,75
7	8	0	0	0,875
8	1	0	0	1

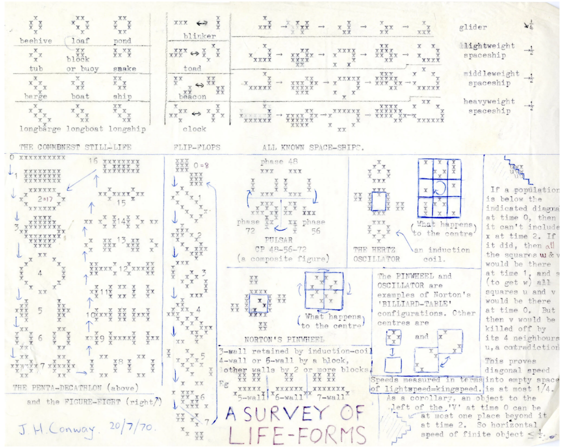
Wolfram's classification and normalized Langdon's λ metric



Conway's "Game of Life"

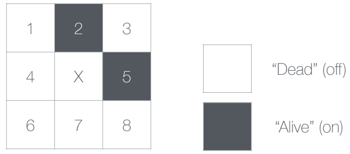
- 2-Dimensional Cellular Automata
- Developed by British mathematician John Conway
- Made famous by Martin Gardner in his "Mathematical Games" column in Scientific American of October 1970

Conway's "Game of Life"



Conway's "Game of Life"

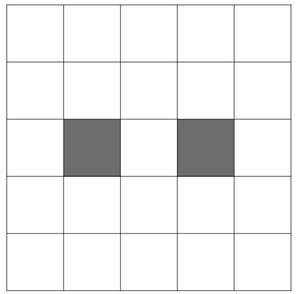
- Each cell (on an infinite plane) has eight neighbors
- Each cell can be "alive" or "dead"
- Cells come alive, die or survive according to simple rules



Conway's "Game of Life"

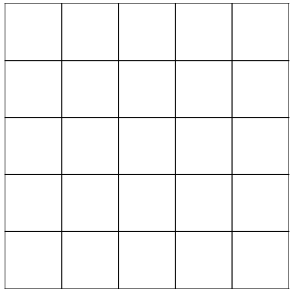
- Rules:
  - a live cell with 2 or 3 live neighbors survives (survival)
  - a live cell with fewer than 2 live neighbors dies (death from loneliness)
  - a live cell with more than 3 live neighbors dies (death from over crowding)
  - a dead cell with exactly 3 live neighbors come alive (birth)

"Game of Life" behaviors



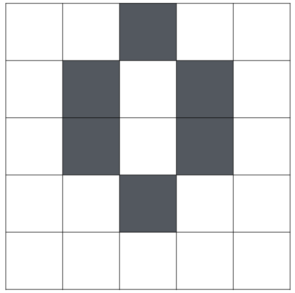
“Game of Life” behaviors

Fixed point



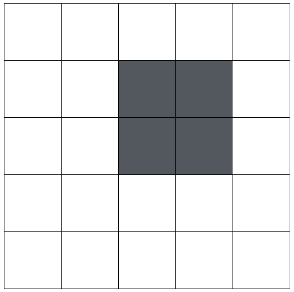
“Game of Life” behaviors

Fixed point

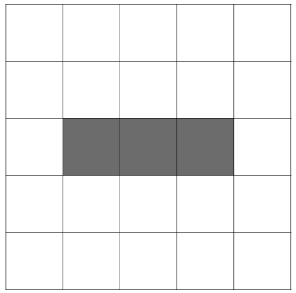


“Game of Life” behaviors

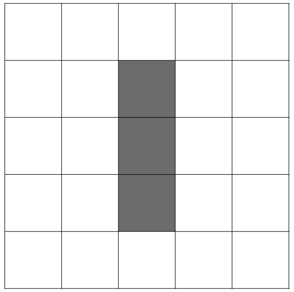
Fixed point



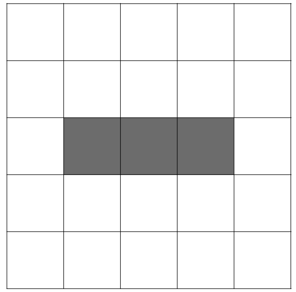
“Game of Life” behaviors



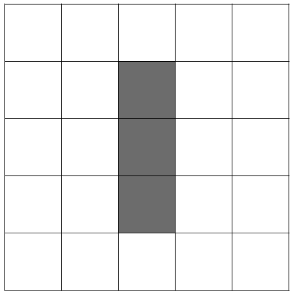
“Game of Life” behaviors



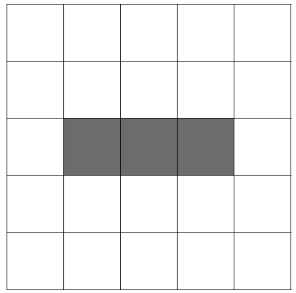
“Game of Life” behaviors



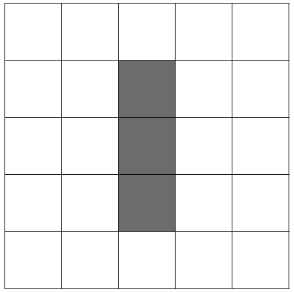
“Game of Life” behaviors



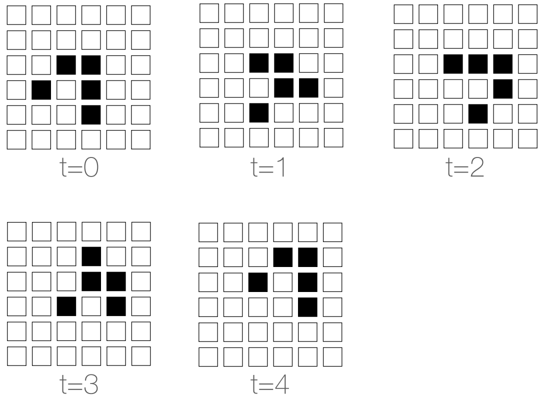
“Game of Life” behaviors



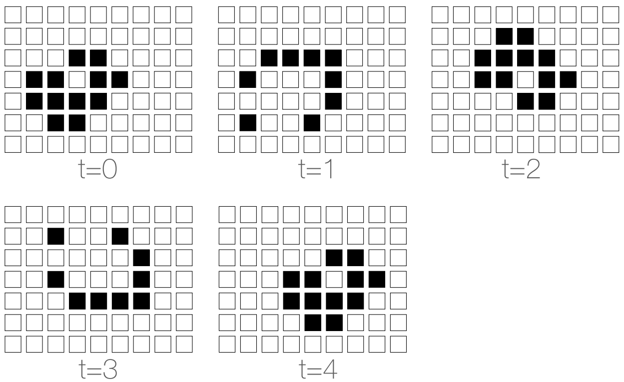
"Game of Life" behaviors



Glider



Fish



- NetLogo GameofLife, Mini-Life

CAs as computers

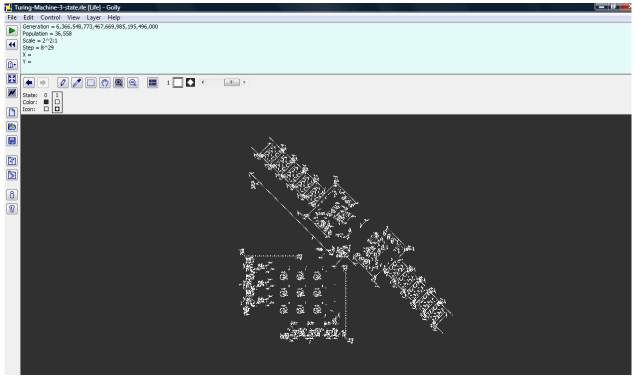
- CAs are capable of performing "computation"
- Computation is the "processing of information" and consists of
  - representing and inputing,
  - storing,
  - transferring,
  - transforming (processing),
  - outputting information
- "Universal computation" — ability to compute anything that is computable
- "Programmable computers" (with infinite resources) are capable of universal computation

CA and universal computation

- Both Conway's game of life and CA rule 110 are capable of universal computation
- Prove by showing that
  - Game of Life or CA 110 are equivalent to a Turing Machine
  - basic logical operators needed for universal computation can be constructed using Game of Life or CA 110

CA and universal computation

- Game of life is equivalent to a Turing Machine (by construction)



Logical operators from "Game of Life"

- Construct basic logical operators using Game of Life
- Building blocks:

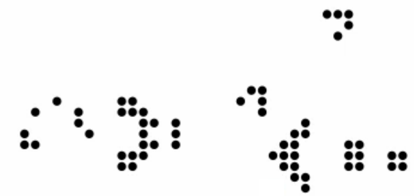
△ - Glider or Fish Gun

▭ - Glider or Fish Eater

● - Data Stream

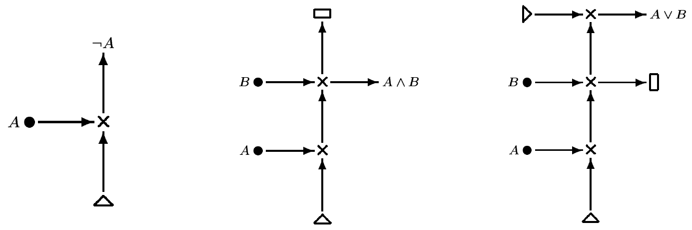
✕ - Collision

Glider gun



- NetLogo GameOfLife

Logical operators from “Game of Life”



NOT AND OR

CA and universal computation

- CAs as “universal computers” are not practical
- Yet, CAs have been used to perform special-purpose, practical parallel computations such as image processing

CA and complex systems

- Idealized models that are capable of producing complex behavior from very simple rules
- Natural complex systems can be modeled using CAs
- CAs allow us to understand how complex dynamics can produce collective “information processing” in a decentralized system