Lecture 7: Cellular Automata

Modelling Complex Systems (Spring 2026)

Jordi-Lluís Figueras

Matematiska Institutionen, Uppsala Universitet

April 17, 2026

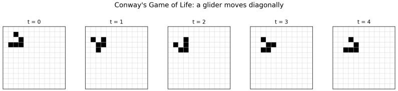
Prelecture discussion

Before Lecture 7, experiment with `gameOfLifeInteractive.py` and `elementaryCaInteractive.py`.

- Draw a few small Life patterns by hand and see which ones die, freeze, or oscillate.
- Try perturbing a stable-looking pattern by changing one cell.
- In the elementary automaton script, compare Rules 8, 50, 30, and 110 from the same initial row.

Questions to reflect on:

- Which Life patterns look fixed, periodic, or mobile?
- Which one-dimensional rules simplify quickly, and which keep generating visible structure?
- How much of the global pattern is already encoded in the local rule?



Lecture goals

In this lecture we will:

- begin with an exploratory discussion of Conway's Game of Life;
- explain the Life rule and identify still lifes, oscillators, and moving patterns;
- use this exploration to motivate the broader notion of a cellular automaton;
- define cellular automata as discrete dynamical systems with local update rules;
- study elementary cellular automata in one dimension;
- explain how Wolfram's rule numbering produces the 256 elementary rules;
- discuss Wolfram's empirical classification into classes 1–4;
- use cellular automata to illustrate emergence, pattern formation, and computational universality;
- discuss why cellular automata are conceptually powerful but often too abstract for direct quantitative modelling.

Exploring the Game of Life

Start from the example, not the abstraction

Before giving a general definition, it is useful to look at one concrete system. The Game of Life is ideal for this because:

- the local rule is very short;
- the pictures are easy to generate and recognize;
- the long-time behavior can still be surprisingly rich.

So for the first part of the lecture, ask simple observational questions:

- Which patterns die out quickly?
- Which patterns persist forever?
- Which patterns repeat or move?
- Which features seem robust under small perturbations?

Conway's Game of Life

John Horton Conway (1937–2020)

Source: MacTutor History of Mathematics Archive

Conway's Game of Life is played on:

- a two-dimensional square grid;
- with two states, *dead* (0) and *alive* (1);
- using the eight-cell Moore neighborhood.

Moore neighborhood: 8 surrounding cells

If $N(i,j)$ denotes the number of alive neighbors of cell (i,j), the rule is B3/S23: birth for $N(i,j) = 3$, survival for $N(i,j) = 2$ or 3, and death otherwise.

Source: Moore-neighborhood diagram by the lecturer.

Reading the rule qualitatively

The Life rule can be read as a balance between growth and crowding:

- **birth:** exactly three alive neighbors create a new alive cell;
- **survival:** two or three alive neighbors support an existing alive cell;
- **underpopulation:** too few neighbors lead to death;
- **overpopulation:** too many neighbors also lead to death.

This language is suggestive, but the model should be understood as an abstract rule-based system rather than a literal biological model.

What should we look for in the first experiments?

When we start from a hand-made pattern or from a random initial condition, there are several possibilities:

- the configuration may disappear completely;
- it may settle to a fixed shape;
- it may enter a periodic cycle;
- it may emit moving pieces that travel across the grid;
- it may leave behind a complicated mixture of stable and oscillatory fragments.

Even before introducing any formal notation, we already see a familiar modelling question:

Observation

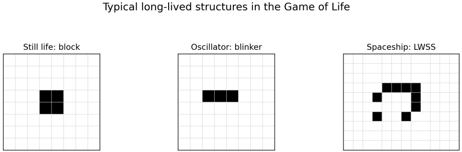
How much of the large-scale behavior can be inferred from the local rule alone?

Still lifes, oscillators, and spaceships

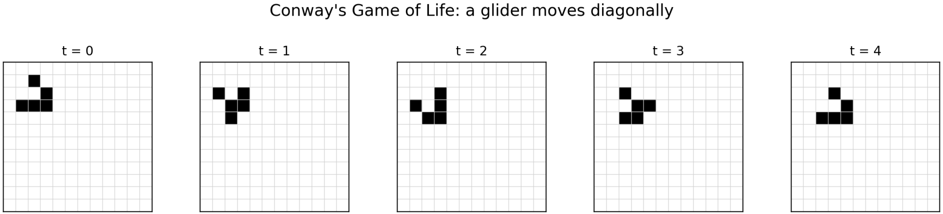
Repeated iteration of the Life rule produces several robust types of structures:

- **still lifes:** patterns that remain unchanged;
- **oscillators:** patterns that repeat after a finite number of steps;
- **spaceships:** patterns that reappear translated in space.

This is already enough to make the system interesting: we see persistence, periodicity, and effective transport, all generated by one local rule.

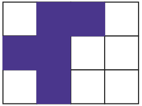


Example: the glider



A glider returns to the same shape after four updates, shifted by one cell diagonally. It is one of the simplest examples showing how local deterministic rules can create effective motion and signal propagation.

Example: a small seed with a large outburst



Initial pattern: the *R*-pentomino

Starting from only five live cells, the *R*-pentomino produces a surprisingly large and irregular transient.

- It is one of the classical examples of a small seed with a large outburst.
- The pattern grows into a messy cloud before eventually settling.
- This is a good classroom example of how tiny local data can generate complicated global evolution.

What the Game of Life already teaches us

Without any formalism yet, the Game of Life already shows that:

- deterministic local rules can generate unexpected global structures;
- simple initial seeds can have very different long-time outcomes;
- repeated local interactions can create moving and interacting objects.

This is the point where it becomes natural to step back and ask for a general framework.

From the Game of Life to cellular automata

What is a cellular automaton?

A cellular automaton is a dynamical system with four basic ingredients:

1. a collection of cells arranged on a lattice or grid;
2. a finite set of states available at each cell;
3. a neighborhood specifying which nearby cells influence an update;
4. a local rule applied synchronously at each discrete time step.

The main idea is simple but powerful:

Local-to-global principle

Complicated population-level patterns may arise from the repeated application of a very simple local rule.

Cellular automata fit our dynamical-systems framework

In previous lectures we studied iterated maps

$$x_{n+1} = F(x_n).$$

A cellular automaton has exactly the same structure, but the state is now a whole configuration of cells.
For a one-dimensional lattice indexed by $i \in \mathbb{Z}$, write the state at time n as

$$x^n = (x_i^n)_{i\in\mathbb{Z}}, \qquad x_i^n \in S,$$

where S is a finite state set, for example $S = \{0, 1\}$.
Then the global evolution is again

$$x^{n+1} = F(x^n),$$

but the map F is built from the same local update rule at every site.

Ingredients of a cellular automaton model

When defining a cellular automaton, one must specify:

- **lattice:** one-dimensional line, two-dimensional square grid, hexagonal grid, etc.;
- **state space:** binary states, several colors, or a finite set of symbols;
- **neighborhood:** nearest neighbors, larger radius, Moore neighborhood, and so on;
- **boundary conditions:** infinite lattice, periodic boundary conditions, or fixed walls;
- **initial condition:** single seed, finite pattern, or random configuration.

These choices strongly affect what structures can appear, even before the rule itself is changed.

Why cellular automata matter in complex systems

Cellular automata are interesting because they separate two levels of description:

- the microscopic rule is local and easy to state;
- the macroscopic pattern can be difficult to predict from inspection alone.

This makes them a clean laboratory for questions such as:

- How does structure emerge from decentralized interactions?
- Which types of rules lead to order, periodicity, or apparent randomness?
- How much complexity is possible without fine-tuning or sophisticated components?

Elementary cellular automata

Elementary cellular automata: the simplest nontrivial case

An elementary cellular automaton has:

- a one-dimensional lattice of cells;
- two states at each site, usually 0 and 1;
- a radius-1 neighborhood consisting of the cell and its immediate left and right neighbors.

So the update rule has the form

$$x_i^{n+1} = f(x_{i-1}^n, x_i^n, x_{i+1}^n), \qquad f : \{0,1\}^3 \to \{0,1\}.$$

Even this extremely small rule space already contains fixed behavior, periodic behavior, irregular behavior, and structured long-lived patterns.

Why there are exactly 256 elementary rules

For a binary radius-1 rule, each local neighborhood is a triple

$$(x_{i-1}^n, x_i^n, x_{i+1}^n) \in \{0,1\}^3.$$

There are exactly

$$2^3 = 8$$

possible neighborhoods:

$$111,\ 110,\ 101,\ 100,\ 011,\ 010,\ 001,\ 000.$$

For each of these 8 inputs, the output can be either 0 or 1. Therefore the total number of different rules is

$$2^8 = 256.$$

Rule encoding: example of Rule 110

The eight outputs are usually read in the order

$$111,\ 110,\ 101,\ 100,\ 011,\ 010,\ 001,\ 000.$$

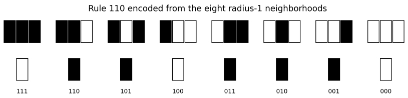
For Rule 110, the outputs are

$$0\ 1\ 1\ 0\ 1\ 1\ 1\ 0,$$

which is the binary number

$$01101110_2 = 110.$$

This coding gives Wolfram's numbering convention for elementary cellular automata.

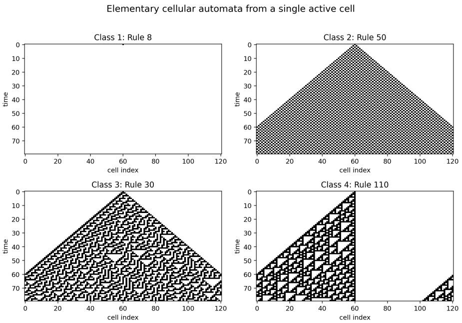


Examples of elementary rules

Starting from one active cell, different rules produce qualitatively different spacetime patterns.

- Rule 8 quickly collapses to a simple state.
- Rule 50 produces repeating motifs.
- Rule 30 generates an irregular triangular pattern.
- Rule 110 supports coherent moving structures.

This is one of the central lessons of cellular automata: qualitative complexity is not proportional to the length of the rule.



Experimenting with the interactive elementary-CA script

Use `elementaryCaInteractive.py` to test how the same initial row evolves under different rules.

- Start from a single active cell and compare Rules 8, 50, 30, and 110.
- Draw a short initial row by hand and see which features survive, spread, or disappear.
- Change only one cell in the top row and observe how strongly the spacetime diagram changes.
- Ask whether the rule looks stabilizing, periodic, irregular, or capable of supporting localized structures.

This is a useful bridge between the formal rule definition and the qualitative classes seen in the plots.

Take-away message

- Cellular automata are discrete dynamical systems defined by local update rules on a lattice.
- Even elementary one-dimensional automata already display fixed, periodic, irregular, and highly structured behavior.
- Wolfram's rule numbering explains why there are exactly 256 elementary binary radius-1 rules.
- Conway's Game of Life is the canonical two-dimensional example of emergence from local interactions.
- Rule 110 and the Game of Life show that very simple rules can support universal computation.
- Cellular automata are powerful conceptual models, but their scientific usefulness depends on how much abstraction the application can tolerate.

Suggested bibliography

For students who want to read more:

- M. Mitchell, *Complexity: A Guided Tour*, Oxford University Press, 2009.
- S. Wolfram, *A New Kind of Science*, Wolfram Media, 2002.
- A. Ilachinski, *Cellular Automata: A Discrete Universe*, World Scientific, 2001.
- E. R. Berlekamp, J. H. Conway, R. K. Guy, *Winning Ways for Your Mathematical Plays*, Vol. 2, A K Peters, 2003.
- J. von Neumann, *Theory of Self-Reproducing Automata*, edited by A. W. Burks, University of Illinois Press, 1966.

Acknowledgment

These slides were prepared with the assistance of OpenAI Codex / ChatGPT 5.4 as a drafting and editing tool.
All mathematical, pedagogical, and course-specific content remains under the responsibility of the lecturer.