Agenda: Complex Processes in Nature and Laboratory

Order and Chaos, determinism and unpredictability

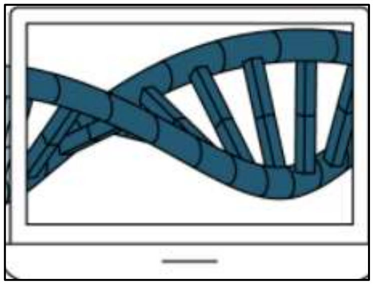
Non-linear dynamics in nature and their modeling
- Examples (climate, planetary motion),
- mathematical model (logistic map)
- Stability criteria, stationary states

Complex kinetics in coupled chemical reactions
- Self-organization in coupled chemical reactions
- Application: Self-replication in autocatalytic reactions
- Cellular automata and fractal structures

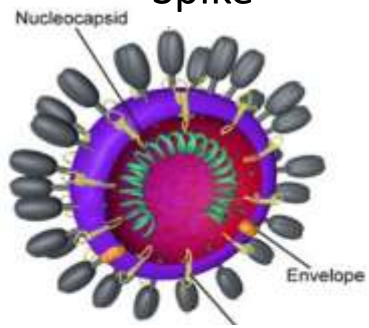
Thermodynamic states and their transformations
- Collective and chaotic multi-dimensional systems
- Energy types equilibration,
- flow of heat and radiation

Application Vaccine Design

Computer DNA design



Spike

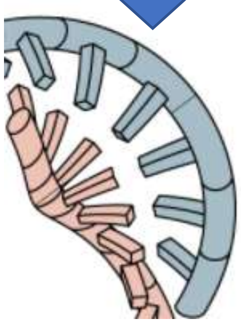


Belouzard et al., 2012)

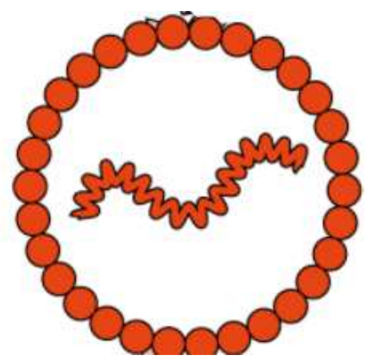
Genetic sequencing Covi-19 virus and mutants, identify specific sub-sequence

Synthetic DNA design for building instructions for Sars-CoV-2 spike protein = one part of virus shell, not a pathogen, but generates antibodies

Insert synthetic DNA as a “template” into bacteria, mass breed bacteria → produce large mass of spike DNA by replication, harvest from bacteria



Separate DNA strands → use enzyme RNA polymerase to make mRNA

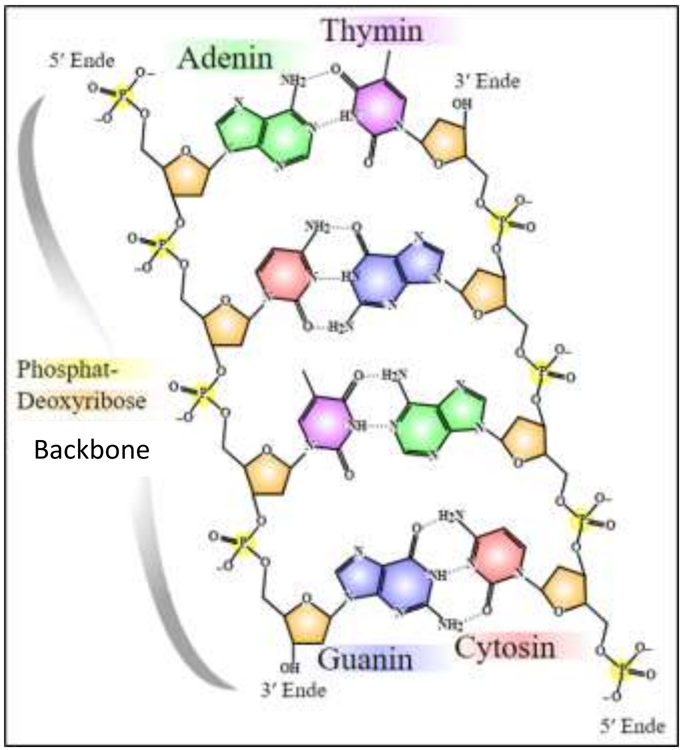


Insert into lipid nanoparticle, safe transfer into body

DNA/RNA Building Blocks

James Watson and Francis Crick (1951) structure of DNA:
Genetic code = encrypted functionality and structure of biologic tissue matter.

DNA Structure



Excerpt genetic sequence

—G-G-C-G-C-C—

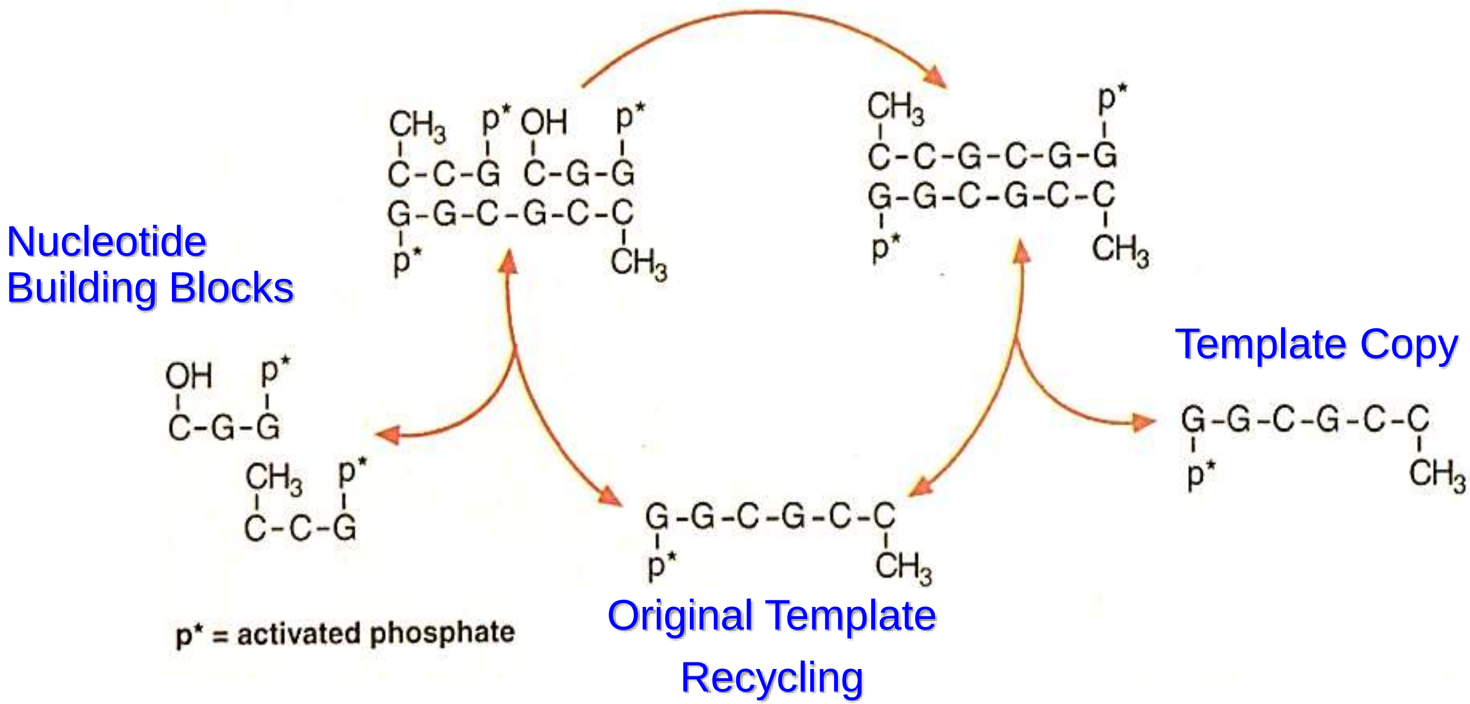
Main DNA components =
Bases: adenine (A), cytosine (C), guanine (G), and thymine (T)

Bases can form pairs and link together → DNA double helix

	DNA	RNA
Shape	Double helix	Single strand
Sugar	Deoxyribose	Ribose
Base	Thymine	Uracil
Length	Long	Short

Complementary Base-Pairing

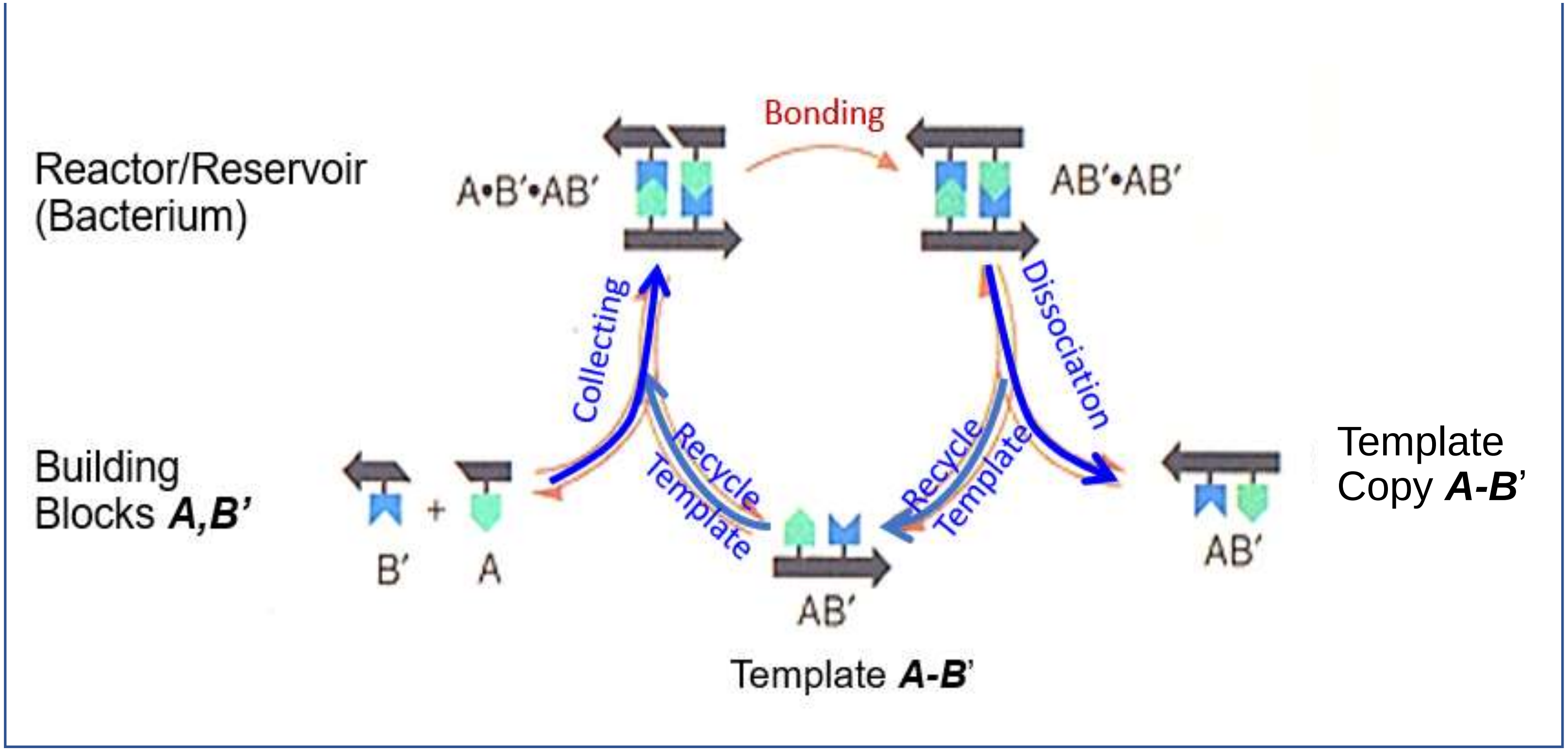
DNA template = palindromic (left-right self-complementary)



Use 2 nucleotide trimers to align and pair with hexa-DNA template, bind the two trimers → bound hexamer on template → Recover original template plus one copy.

Autocatalytic Self-Replication

Autocatalytic self-replication with template: Cycle combines available separate building blocks *A, B'* on template, dissociates template from copy, Re-cycles template



Each cycle makes another template copy → exponential growth in numbers, inhibits other competing processes.

Mathematics of Self-Replication

Represent replication structure by set of rules:

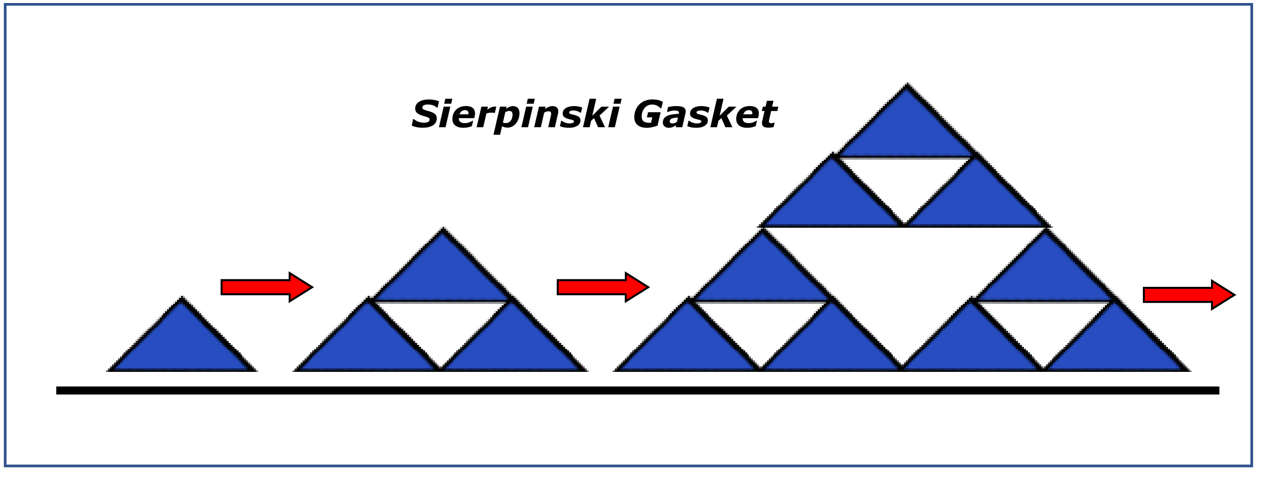
Simples case rule → function ***G***,

"Parent" = Object $\boldsymbol{x = x_0}$ starting point of a series

Descendant ("child") $\boldsymbol{x_1 = G(x_0)}$ = copy of $\boldsymbol{x_0}$,

possible transformation (scaling, translation, …) → Self-Similar Structures

$$x_n = \mathbf{G}(x_{n-1}) = \mathbf{G}[\mathbf{G}(x_{n-2})] = \mathbf{G}\{\mathbf{G}[\mathbf{G}(x_{n-3})]\} = \mathbf{G}^n(x_0) \quad \rightarrow \textbf{\textit{Map}}$$

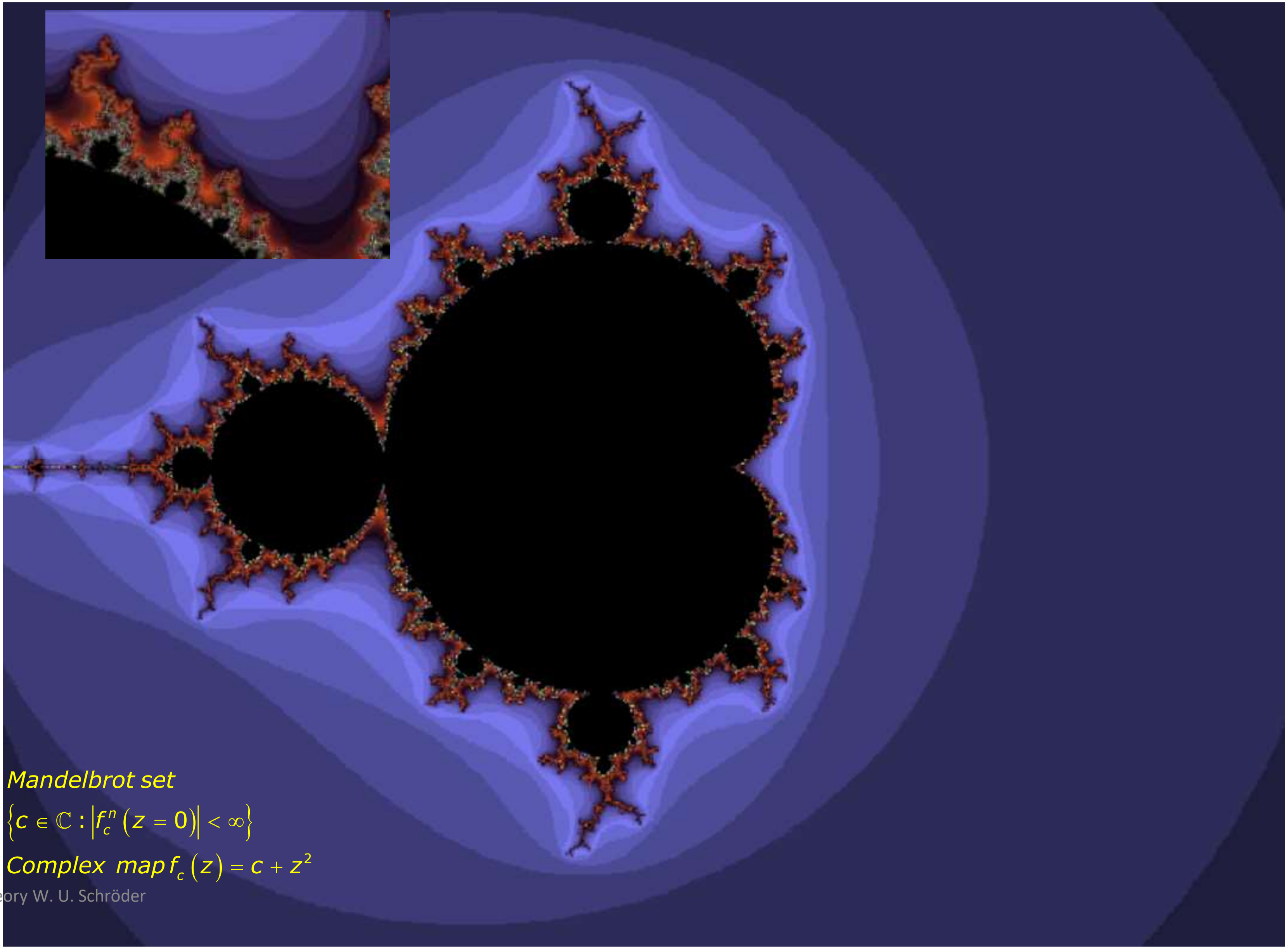


Generation $\boldsymbol{x_0}$ $\boldsymbol{x_1 = G(x_0)}$ $\boldsymbol{G(x_1)}$

Fractal structures

See tutorial

Fractal Mandelbrot Set



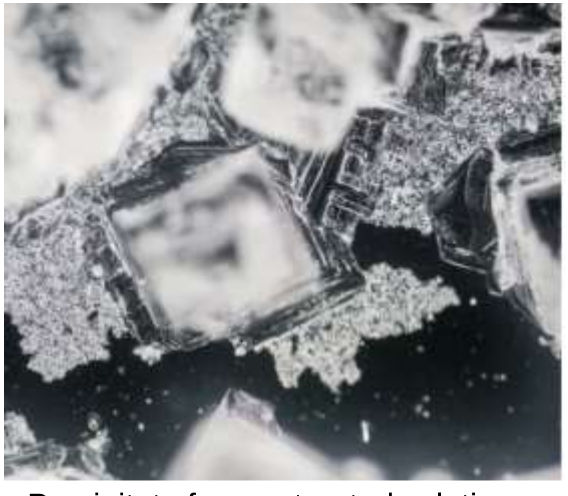
Mandelbrot set

$$\left\{ c \in \mathbb{C} : \left| f_c^n \left(z = 0 \right) \right| < \infty \right\}$$

Complex map $f_c\left(z\right) = c + z^2$

Simple Rules Govern Complexity

Manganese dendrites on limestone



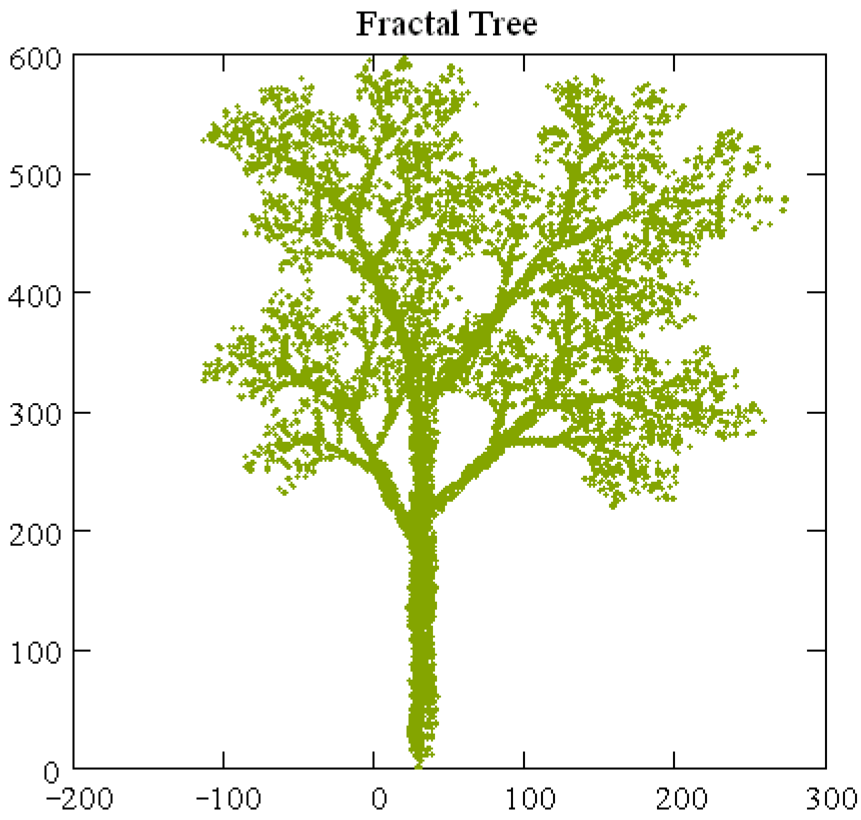
Precipitate from saturated solution

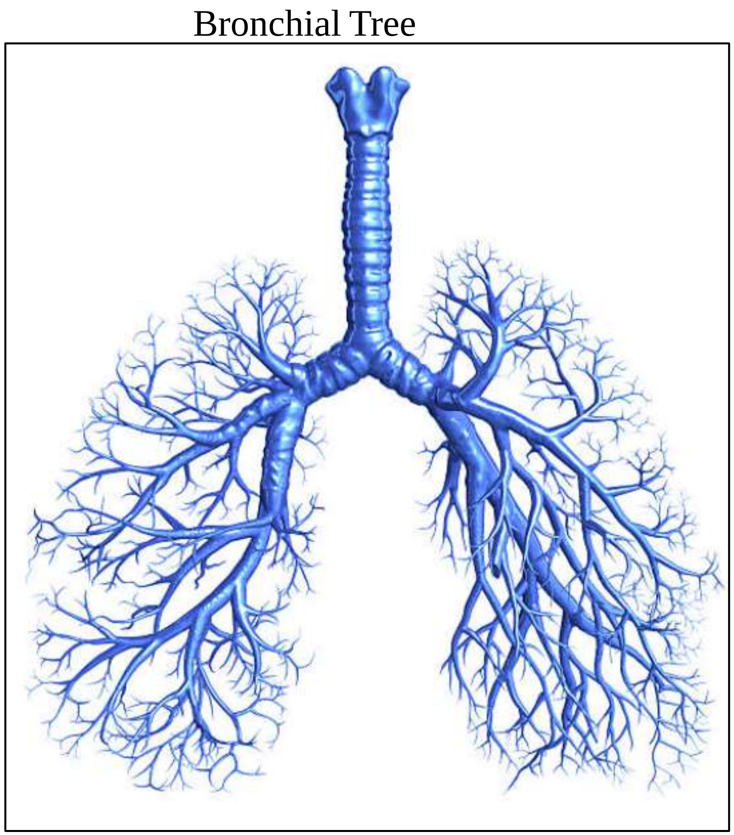
..in processes and structures to which they lead

- self-organizing, orderly behavior,
- co-operative phenomena
  global fractal structures,
- In the limit: complete disorder/chaos

Examples:

- Behavior of gas and liquids, diffusion, convection
- Crystallization
- Turbulence in fluids
- biological life (stem cells), ageing
- Forest fire propagation
- The flow of electricity in a power grid
- Urban development.
- .
- .


Fractal Tree
600
500
400
300
200
100
0
-200
-100
0
100
200
300


Bronchial Tree

Agenda: Complex Processes in Nature and Laboratory

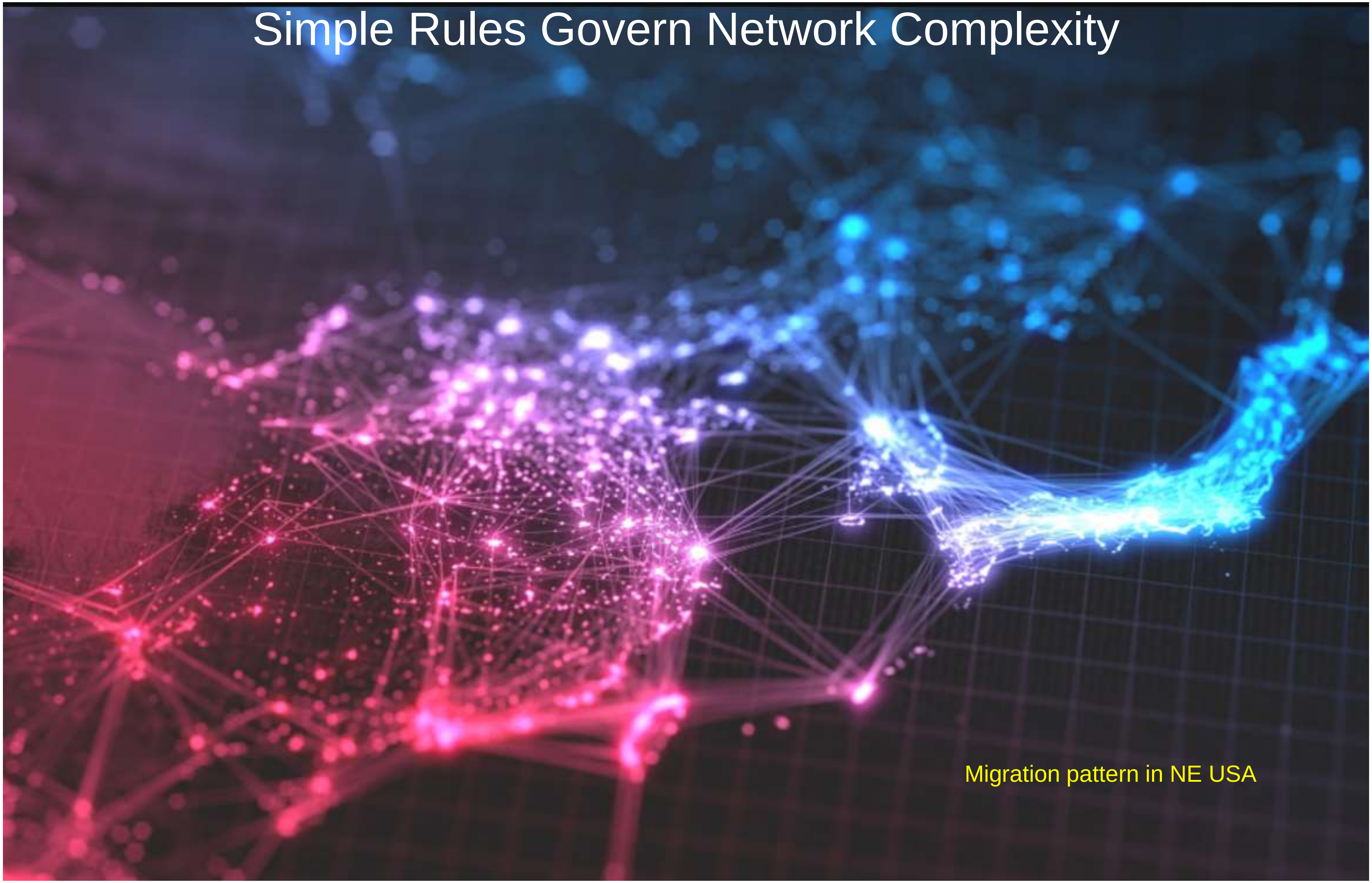
Order and Chaos, determinism and unpredictability

Non-linear dynamics in nature and their modeling
- Examples (climate, planetary motion),
- mathematical model (logistic map)
- Stability criteria, stationary states

Complex kinetics in coupled chemical reactions
- Self-organization in coupled chemical reactions
- Application: Self-replication in autocatalytic reactions
- Cellular automata and fractal structures

Thermodynamic states and their transformations
- Collective and chaotic multi-dimensional systems
- Energy types equilibration,
- flow of heat and radiation

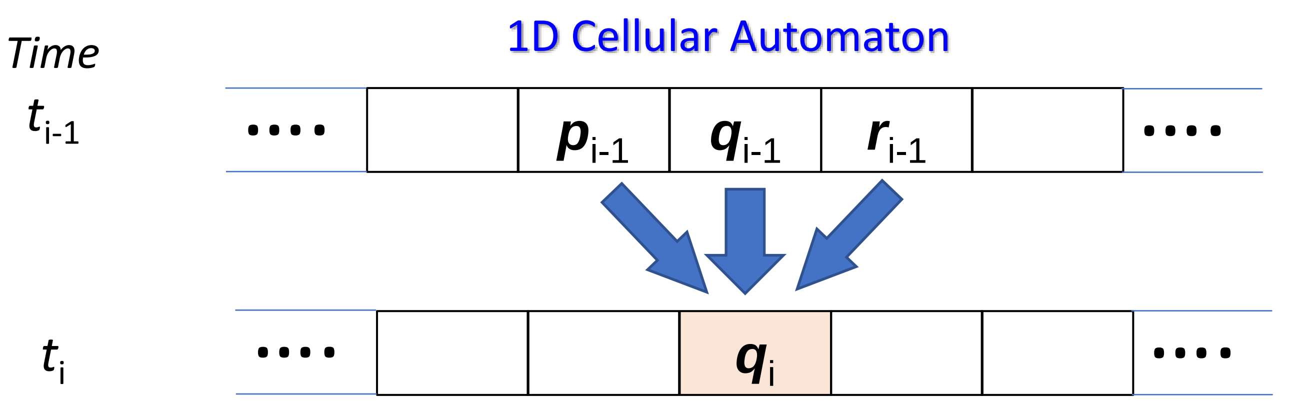
Simple Rules Govern Network Complexity



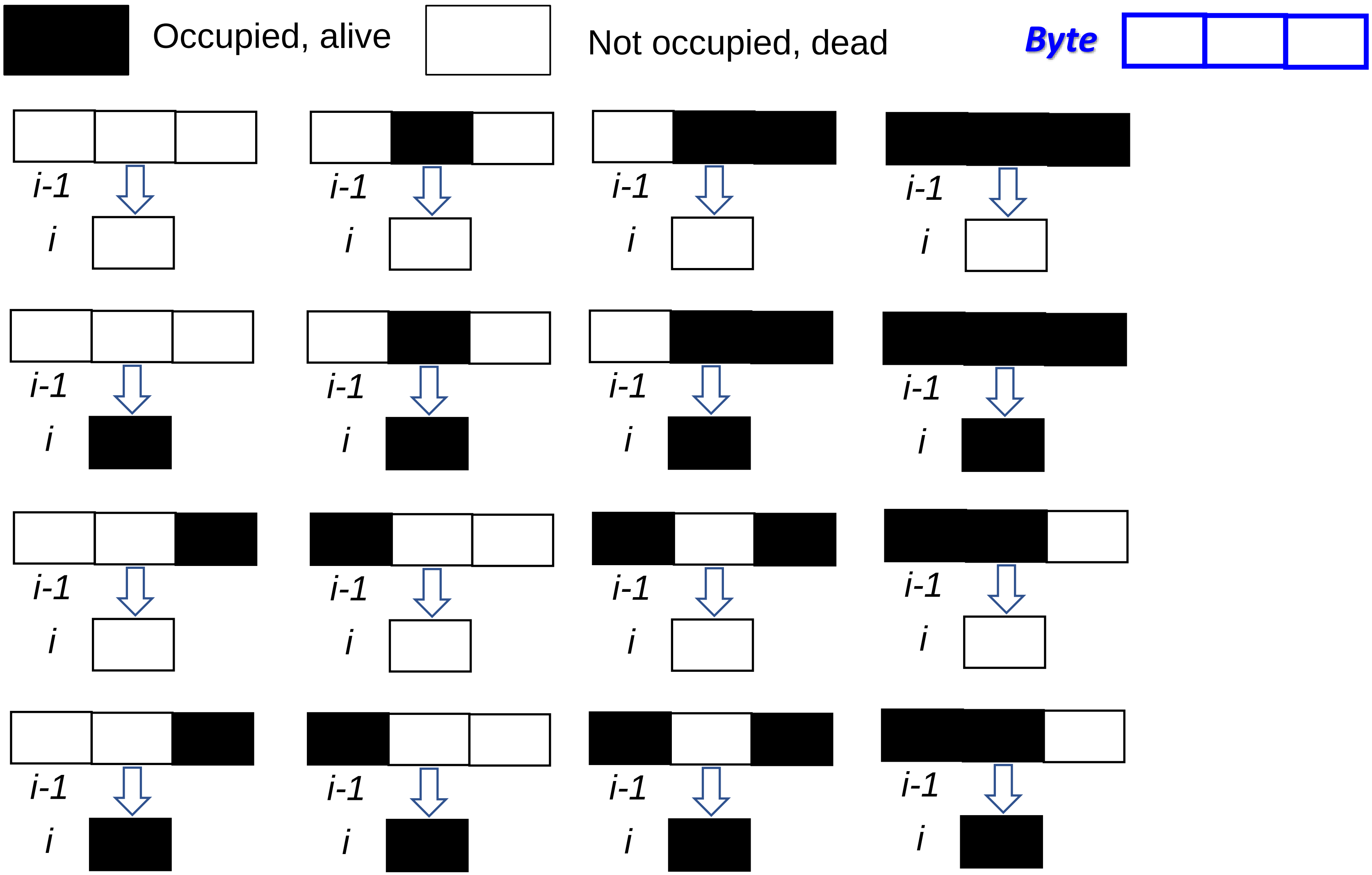
CA Concept

- The configuration (state) space of a system is approximated by an n-dimensional lattice of equal cells.
- Each cell has a finite number of discrete properties.
- Time evolution of CA system occurs (can be modeled) in discrete time steps → generations.
- Evolution occurs to (a set of) strict deterministic rules.
- Evolution rules reference exclusively states of neighboring cells,

  reflect local environment.



Classification of Propagation Rules

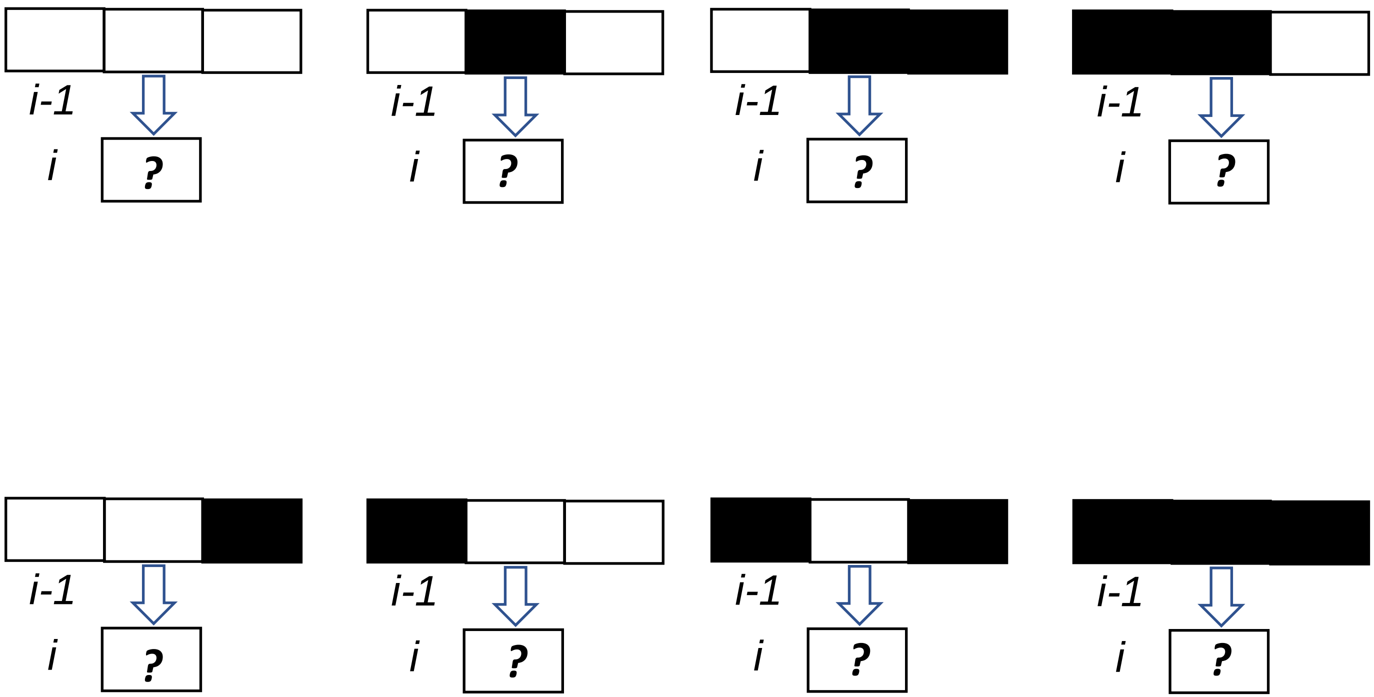


Classification of Propagation Rules

Occupied, alive   Not occupied, dead   ***Byte***

Consolidate:

Any of the patterns could result in the future middle cell to be *alive or dead*

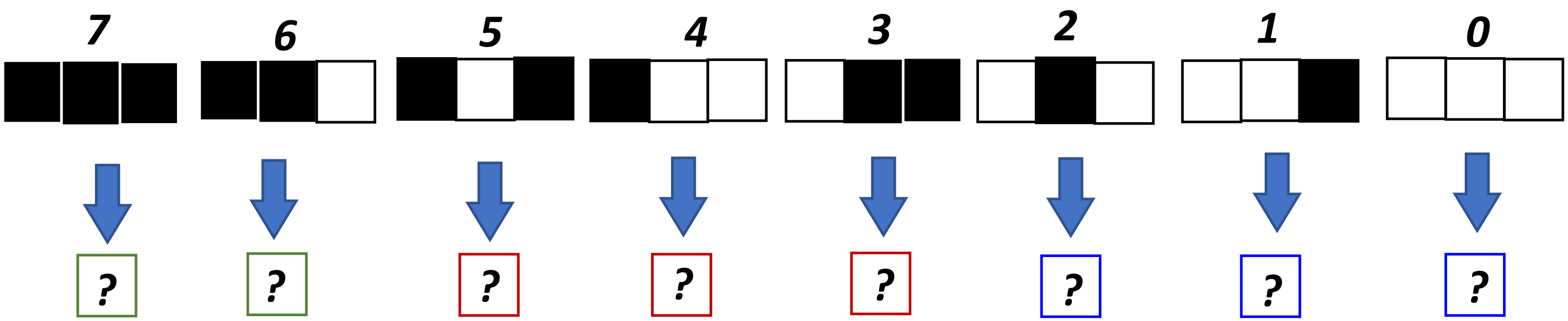


Classification of Propagation Rules

Occupied, alive   Not occupied, dead   ***Byte***

Rearrange byte pattern in ascending order: bytes ordered in binary sequence

Condition of survival of a cell depends on the states of its own past and that of its two neighbors' past → depends on the past state of a triplet of 3 cells (=byte).



There are *8* possible occupation schemes (0, 1, 2, 3, 4, 5, 6, 7) for a byte.
Any one of them, and ***any logically valid combination***, could produce an alive (= ***1***) cell or a dead (= ***0***) cell.
There are obviously 256 (0-255) possible constructions of combination rules (represented by the set of all numbers *0,….,255*).

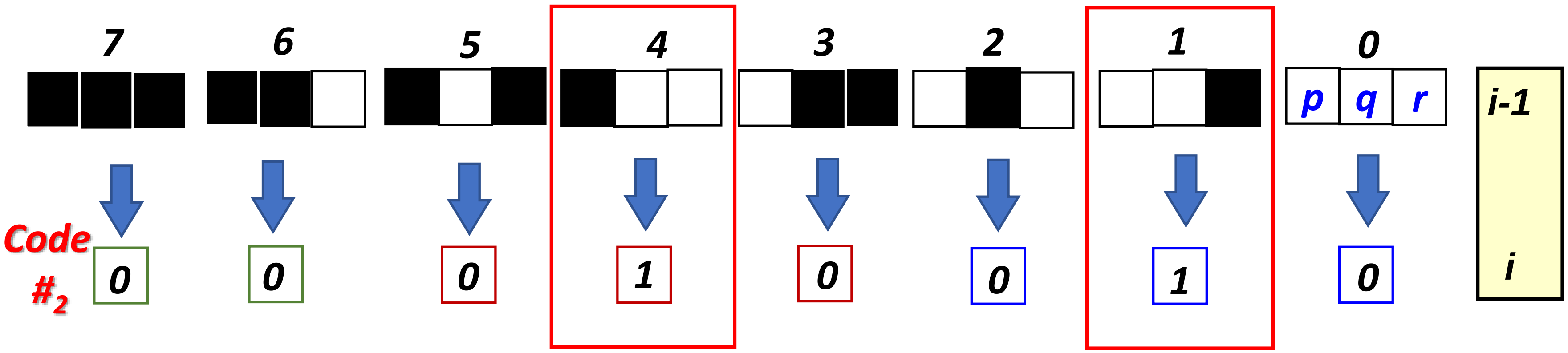
Classification of Propagation Rules

1	Occupied, alive	**0**	Not occupied, dead

Byte

Rearrange byte pattern in ascending order: bytes ordered in binary sequence

Condition of survival of a cell depends on the states of its own past and that of its two neighbors' past → depends on the past state of a triplet of 3 cells (=byte).



Implies live for the cell in the next time step, if it the cell was **previously unoccupied and** had only one alive neighbor on the left or on one on the right, **but not on both sides**.

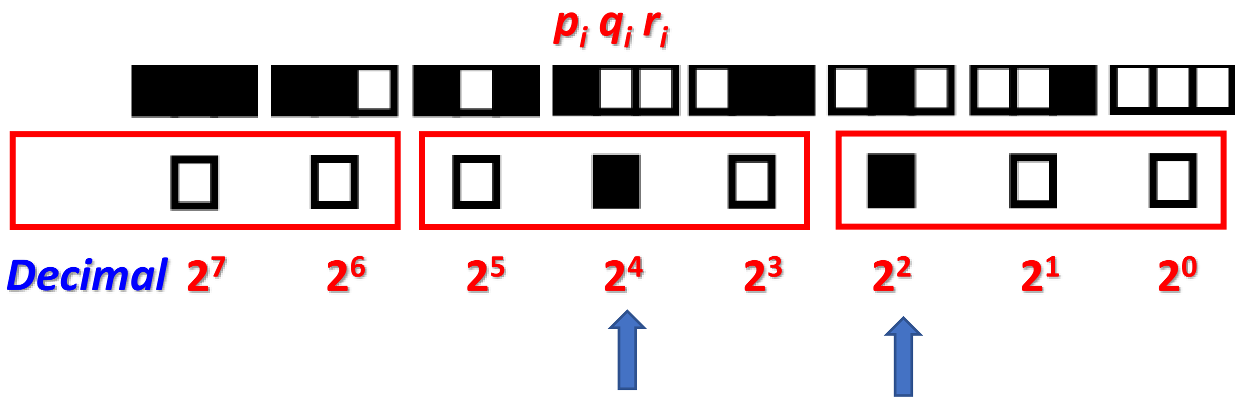
Propagation pattern

$$Code\ \#_{binary} = 10010_2 = (2^4 + 2^1) = 18_{10}$$

$$\rightarrow \boxed{q_i = (p_{i-1} \oplus r_{i-1}) \wedge (\neg q_{i-1})}$$

Interpretation of Propagation Rules

CA code # = $10100_2 = (2^4+2^2) = 20_{10}$



if (p_i) is occupied and (q_i) is not and (r_i) is not

if (q_i) is occupied and (p_i) is not and (r_i) is not

In the next time step, center cell **(q_{i+1})** is populated ⟺

Formal logic

$$q_{i+1} = q_i \wedge \neg(p_i \vee r_i)$$

Truth Table $(p \vee r)$

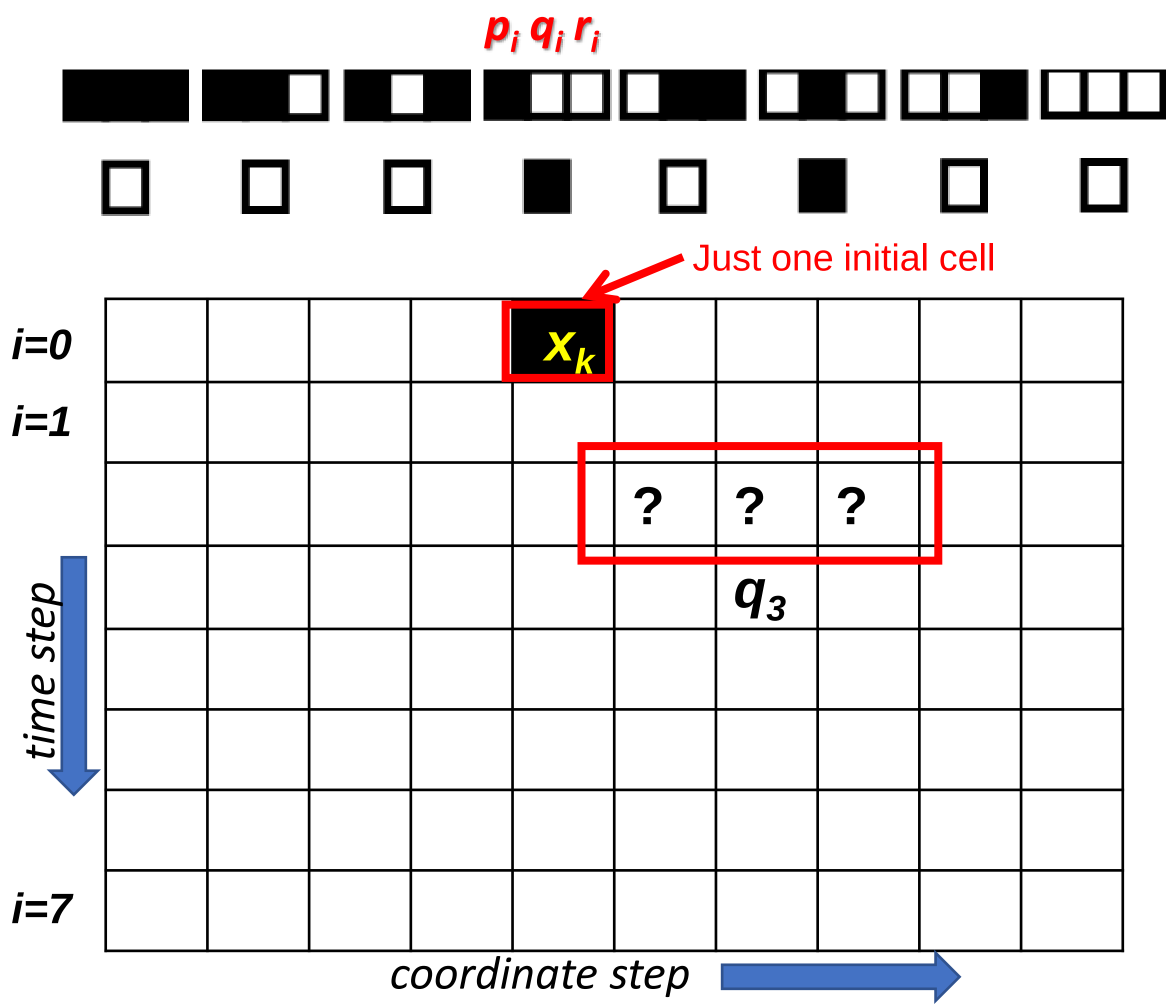
$p \backslash r$	T	F
T	T	T
F	T	F

Constructing the Propagation: CA ***20***

CA code # = $10100_2 = (2^4+2^2) = 20_{10}$

Procedure

1. Draw grid ***x (k)*** vs. ***time (i)***
2. Load initial conditions, pattern $\boldsymbol{x_k}(i=0)$
3. Derive pattern $\boldsymbol{x_k}(i=1)$ from population of triplet $\{\boldsymbol{x_{k-1}}, \boldsymbol{x_k}, \boldsymbol{x_{k+1}},\}$ at (***i=0***)
4. Next row ***i***……
5. For self-replication, keep history

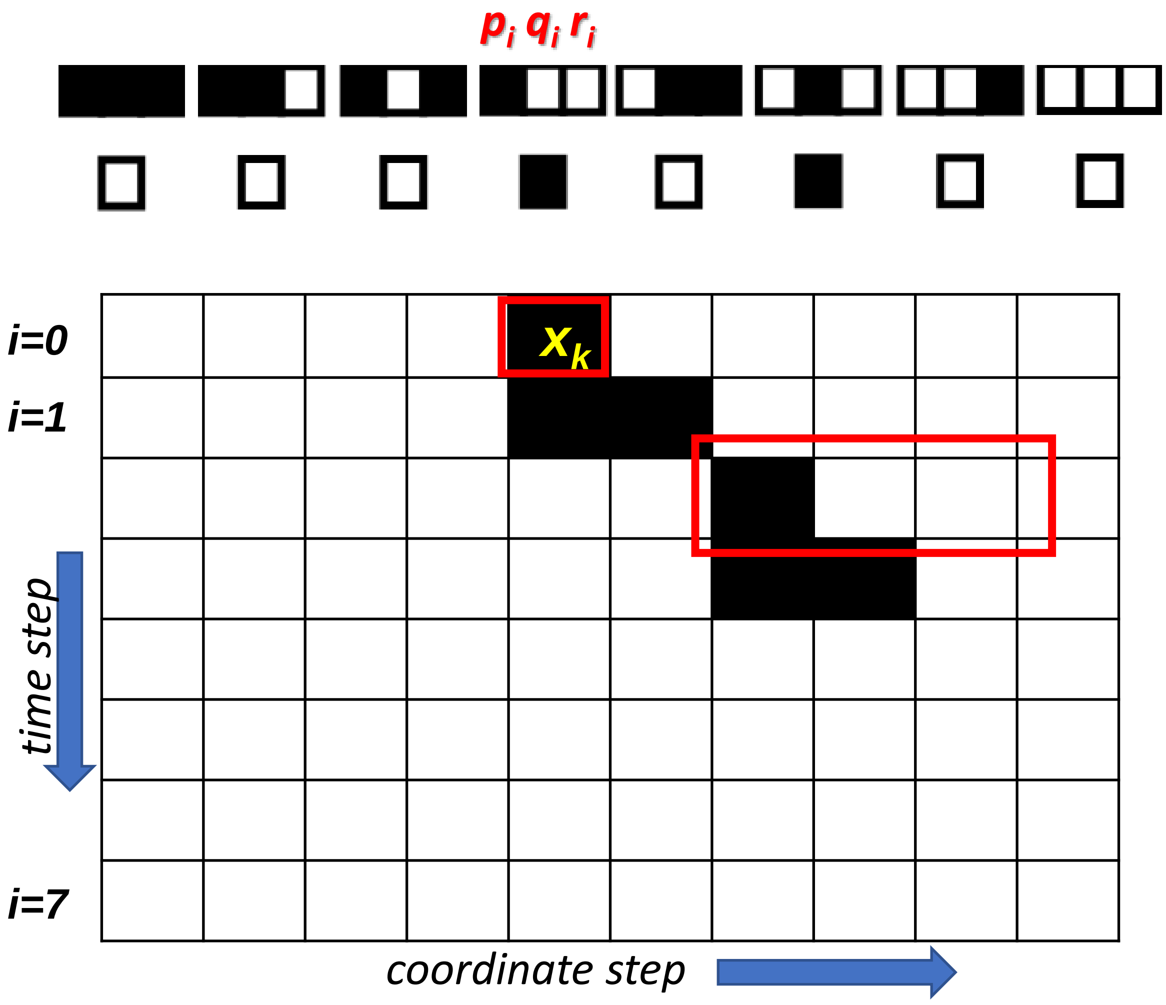


Constructing the Propagation: CA *20*

CA code # = $10100_2 = (2^4+2^2) = 20_{10}$

Procedure

1. Draw grid
   x (k) vs. ***time (i)***
2. Load initial conditions, pattern $x_k(i=0)$
3. Derive pattern $x_k(i=1)$ from population of triplet $\{x_{k-1}, x_k, x_{k+1},\}$ at ($i=0$)
4. Next row ***i*……**
5. For self-replication, keep history

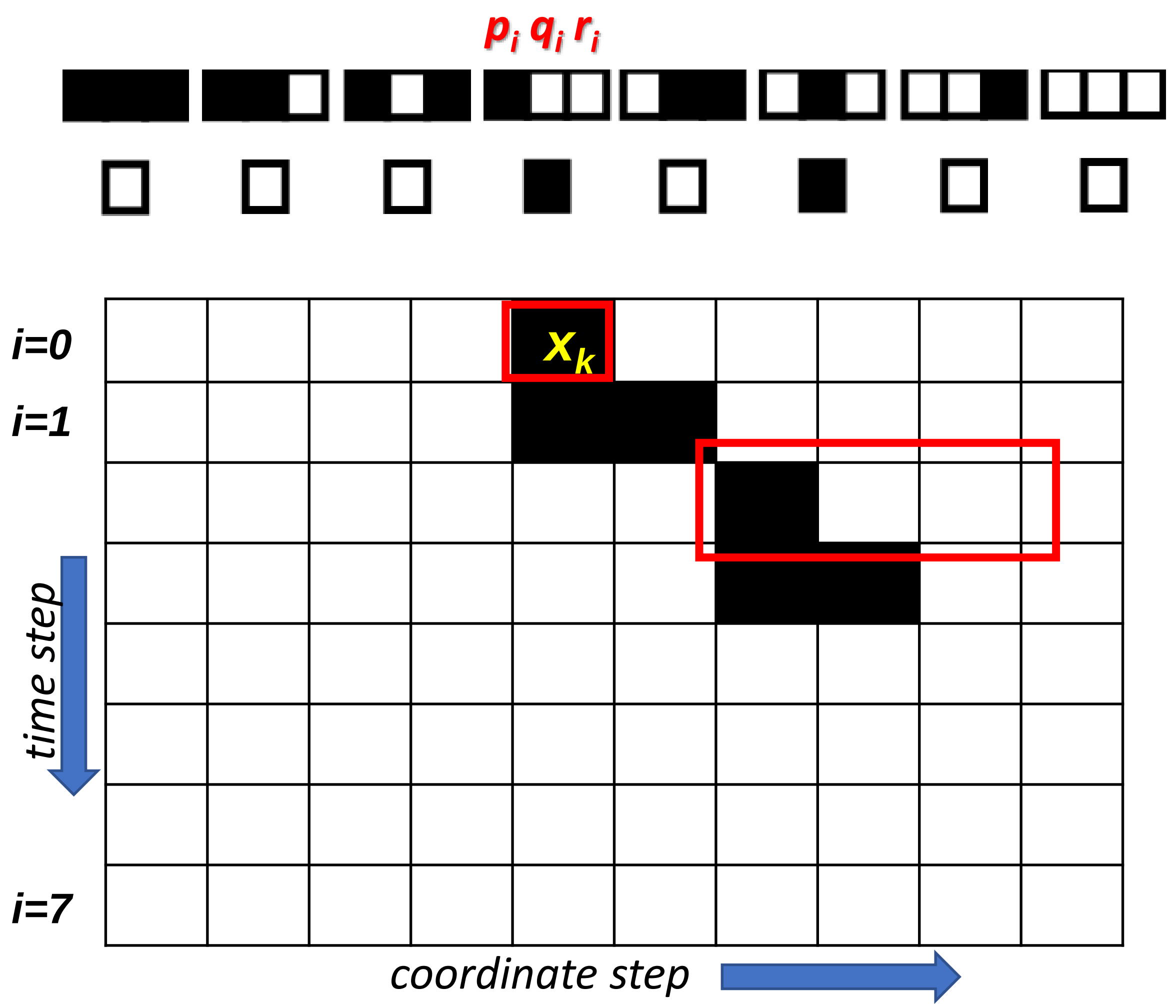


Constructing the Propagation: CA *20*

$CA\ code\ \# = 10100_2 = (2^4+2^2) = 20_{10}$

Procedure

1. Draw grid
   x (k) vs. ***time (i)***
2. Load initial conditions, pattern $\boldsymbol{x}_k(i=0)$
3. Derive pattern $\boldsymbol{x}_k(i=1)$ from population of triplet $\{\boldsymbol{x}_{k-1}, \boldsymbol{x}_k, \boldsymbol{x}_{k+1},\}$ at ($\boldsymbol{i=0}$)
4. Next row ***i***……
5. For self-replication, keep history



Coding CA#20

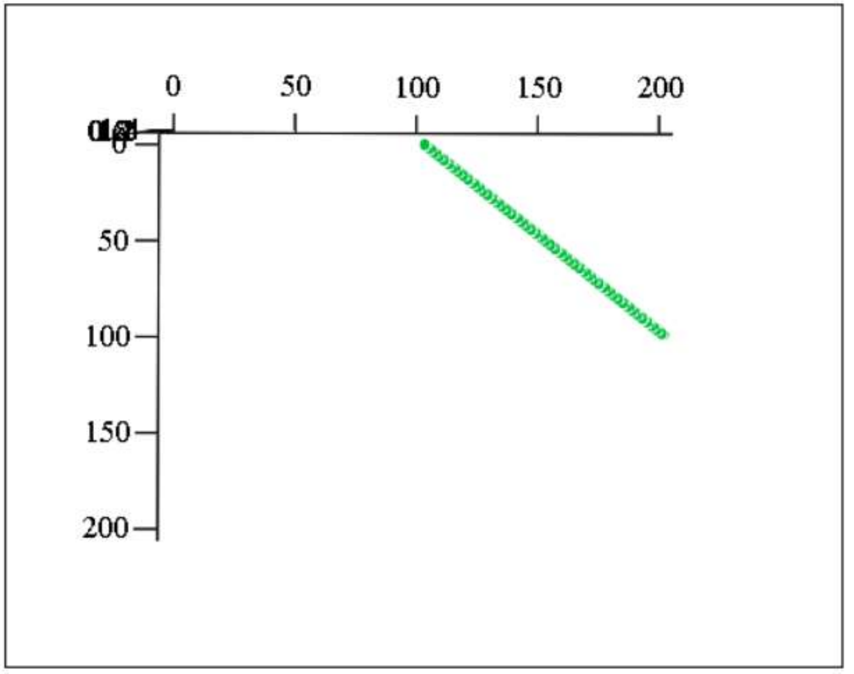
Data: Either single valued initial condition or random with different intensity chosen by width of acceptance window for r
Rule numbers according to Stephen Wolfram (A New
Science , Wolfram Media, 2002

$N := 200$ Data: Array size in x (N) and number of it

$M := 200$

$i := 1..N$

$j := 1..M$



$A_{i,j} := 0$ Initial values

$A_{0,100} := 1$

$j := 1..M-1$

Rule 20: (p Xor q) and not r

$$A_{i,j} := \begin{array}{|l} a \leftarrow 0 \\ p \leftarrow A_{i-1,j-1} \\ q \leftarrow A_{i-1,j} \\ r \leftarrow A_{i-1,j+1} \\ a \leftarrow 1 \;\text{ if } (p = 1 \oplus q = 1) \wedge [\neg(r = 1)] \\ a \end{array}$$

Coding CA#30

$A_{i,j} := 0$

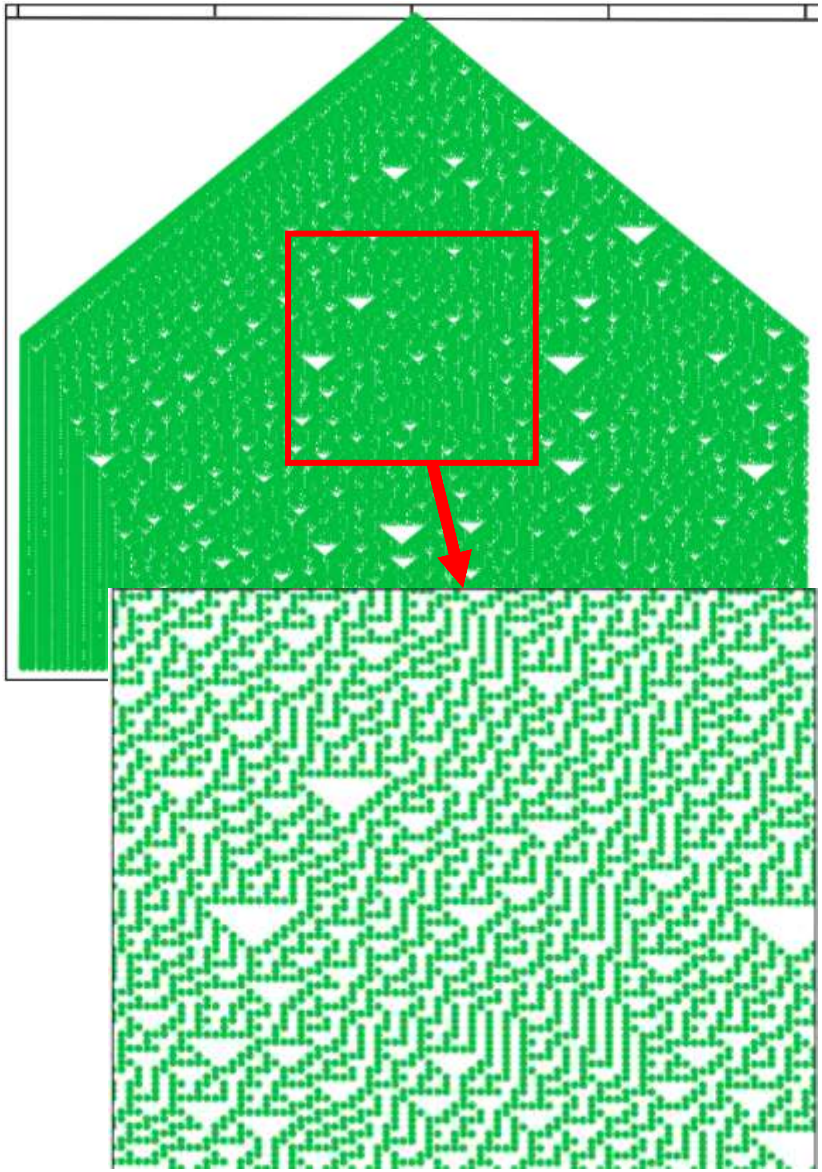
$A_{0,100} := 1$

Initial values 1,0 +

$j := 1..M-1$

Rule 30: p Xor (q or r)
random

$$A_{i,j} := \left| \begin{array}{l} a \leftarrow 0 \\ p \leftarrow A_{i-1,j-1} \\ q \leftarrow A_{i-1,j} \\ r \leftarrow A_{i-1,j+1} \\ a \leftarrow 1 \quad \text{if } p = 1 \oplus (q = 1 \vee r = 1) \\ a \end{array} \right.$$



CA #30, with one more condition. More complex pattern. Repetitive fine structure is observed at the rim of the triangle and upon blowup.

Coding CA#90 Specific IC

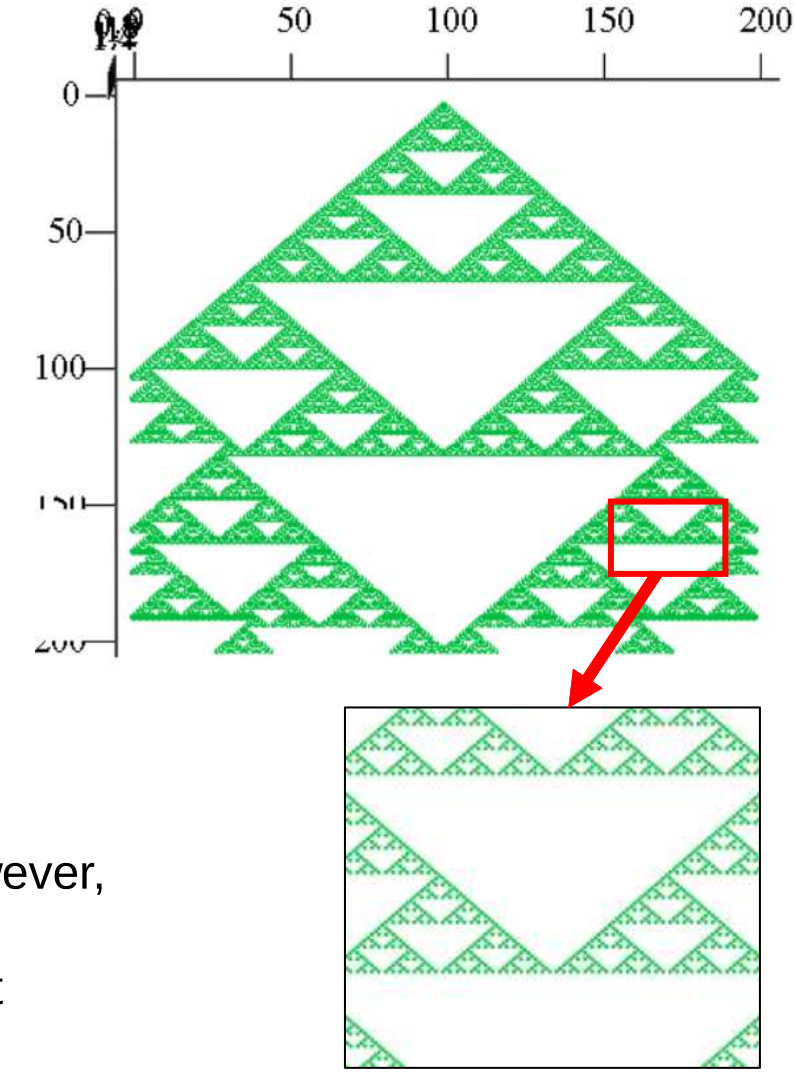
$A_{i,j} := 0$

$A_{0,100} := 1$ Initial values 1,0

$j := 1 .. M - 1$

$$A_{i,j} := \left| \begin{array}{l} a \leftarrow 0 \\ p \leftarrow A_{i-1,j-1} \\ q \leftarrow A_{i-1,j} \\ r \leftarrow A_{i-1,j+1} \\ a \leftarrow 1 \ \ \text{if } (p = 1) \oplus (r = 1) \\ a \end{array} \right.$$

Rule 90: p Xor r
fractal structure



CA #90, expected to show similar randomness as automaton #30. However, there is a highly repetitive pattern of nested triangles. Blow-up: persists at several length scales (fractal).

Coding CA#90 Specific IC

$A_{i,j} := 0$

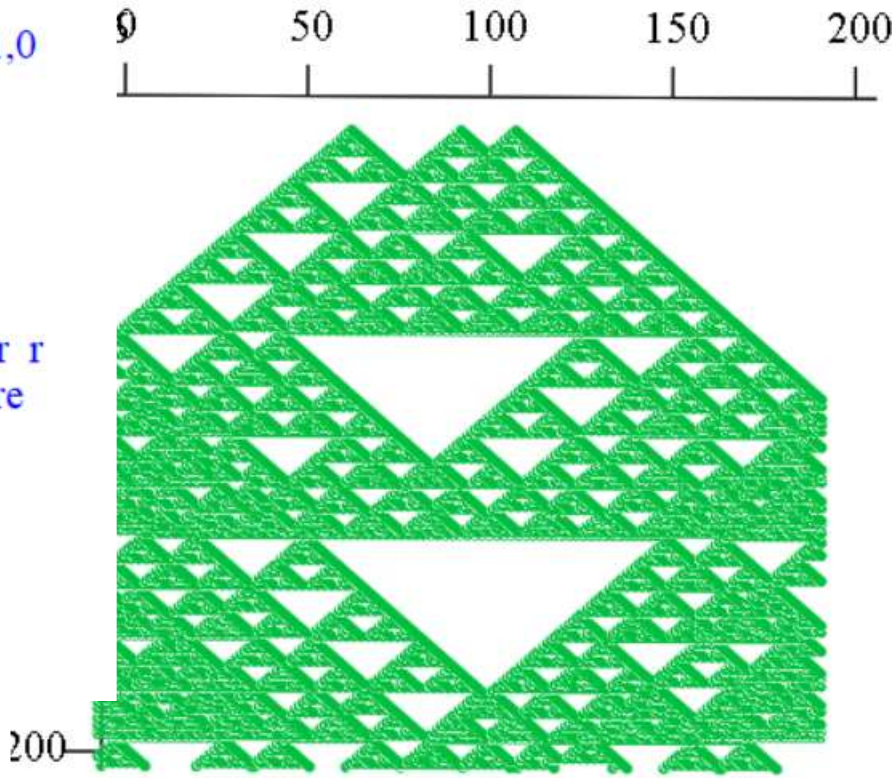
$A_{0,100} := 1$ Initial values 1,0

$A_{0,70} := 1 \quad A_{0,110} := 1$

$j := 1 .. M - 1$

$$A_{i,j} := \begin{array}{|l} a \leftarrow 0 \\ p \leftarrow A_{i-1,j-1} \\ q \leftarrow A_{i-1,j} \\ r \leftarrow A_{i-1,j+1} \\ a \leftarrow 1 \;\; \text{if } (p = 1) \oplus (r = 1) \\ a \end{array}$$

Rule 90: p Xor r
fractal structure



CA #90, with 3 initial cells populated.
Fractal structure is modified but persists.

Coding CA#90 Random IC

$$A_{0,j} := \left|\begin{array}{l} 0 \\ 1 \quad \text{if } \mathrm{rnd}(1) > 0.5 \end{array}\right.$$

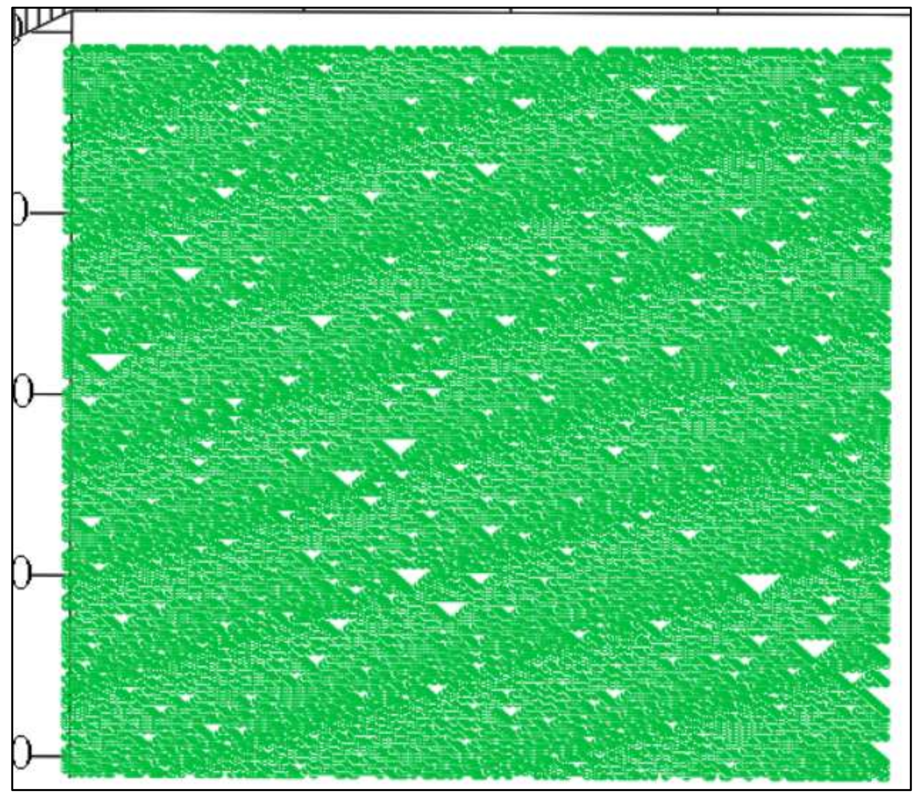
Random initial values 1,0

$$j := 1 .. M - 1$$

$$A_{i,j} := \left|\begin{array}{l} a \leftarrow 0 \\ p \leftarrow A_{i-1,j-1} \\ q \leftarrow A_{i-1,j} \\ r \leftarrow A_{i-1,j+1} \\ a \leftarrow 1 \quad \text{if } (p = 1) \oplus (r = 1) \\ a \end{array}\right.$$

Rule 90: p Xor r
fractal structure

CA #90, with random population of 50% of initial cells.
Specific fractal structure is washed out, additional pattern appears on larger length scale.



Approximates Random chaos

Summary

CA research: 4 classes of automata,

- Class 1 reaches a homogeneous state (all cells free) after a few initial steps.
- Class 2 shows a periodic pattern after the first few steps, relatively independent of initial conditions.
- Class 3 develops into a chaotic pattern, independent of initial conditions.
- Class 4 produces a highly complex, nested fractal pattern.

Very specific, simple, localized microscopic interactions of coupled systems can lead to highly organized structure.

Pronounced fractal structure emerges from localized initial conditions (seeds).

Spread-out initial state conditions lead to washed out structures or chaos.

Because of their specific (unusual) geometrical shape (surface/volume), Class-4

CAs have functionality important for live, biomed and general technology.

Fractal dimensions ➡