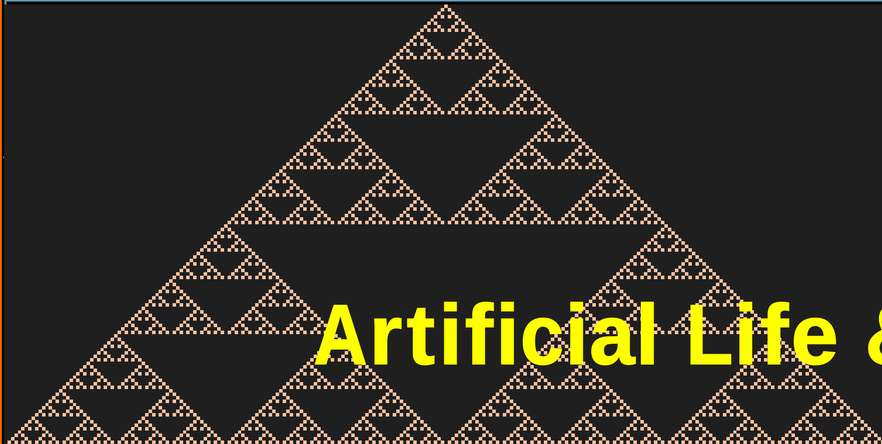


Artificial Life & Complex Systems

Lecture 4

-- Cellular Automata --

May 26, 2007

Max Lungarella

Questions?

Contents

- Basic information
- History
- Examples (1D, 2D)
- Classification

Complex Systems - Properties

Complex Systems – Levels and Scales

- Complex system: Network of interacting objects, agents, elements, or processes that exhibit a dynamic, aggregate behavior
- In a complex system the action of one element affects subsequent actions of other objects in the network, and the action in the whole is more than the simple sum of the actions of its parts
- A complex system is not reducible to a few degrees of freedom or a statistical description

Complex Systems – Levels and Scales

molecular level

Autocatalytic chemical sets

Cellular regulation through gene excitation and inhibition

Multicellular organisms

Collective “super-organisms” such as ant colonies, bee hives, flocks of birds, schools of fish, and oceanic reefs

Larger collections of organisms such as ecosystems, economies, societies

global level

Complex Systems – Synthetic Methodology

Why?

- Top-down analysis too difficult
- Synthesis is the most appropriate approach to the study of complex systems in general and of living complex systems in particular
- "Understanding by building" or synthetic methodology: Start from computational and physical simulations and try to synthesize more and more complex behaviors, which in turn might capture the nature of some aspects of life

Complex Systems and ALife

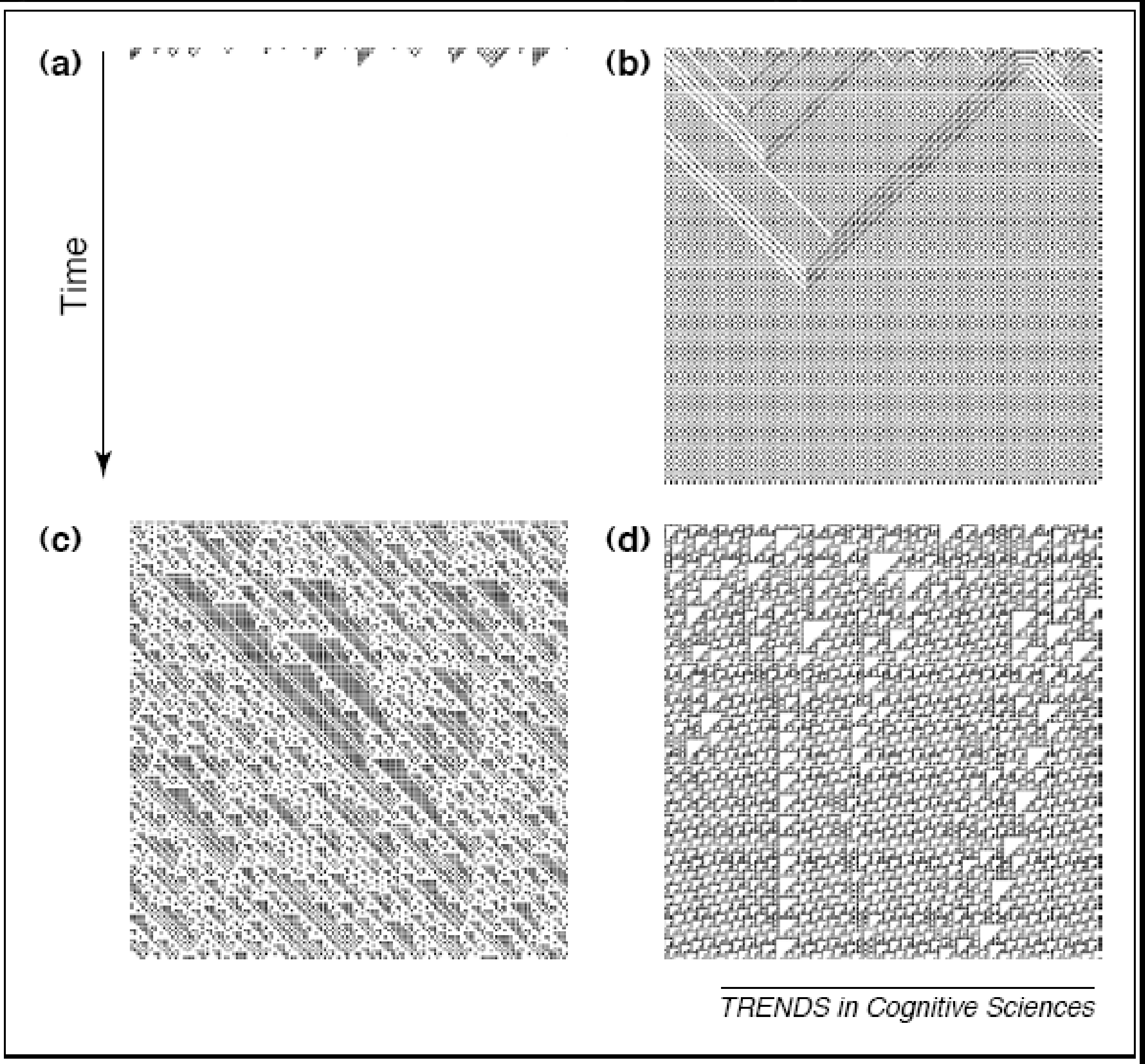
- Synthesis allows to create a much wider spectrum of behaviors: life-as-it-could-be
- Interpretation of the obtained behaviors: Are they life-like? Do they reproduce some aspects of life-as-we-know-it?
- Starting point: what happens if we limit the interaction ability of individuals involved?
  - Do we still get complex behavior?
  - Surprisingly, most simplifications do not affect the ability to get extremely rich and complex behavior at the global level

Cellular Automata: Basic Info

- Cellular automata are mathematical models for complex natural systems containing large numbers of simple identical components with local interactions
- They consist of a spatial lattice of cells, each with a finite set of possible values
- The value of the sites evolve synchronously in discrete time steps according to identical rules (*automata-like*)
- Each cell is a finite-state machine (implementing a set of transition rules) that outputs the next state of the cell, given as input the states of the cells within some finite, local neighborhood

Brief Motivating Example

- Space-time diagrams
- CA = strip of 300 cells each of which is one of two states (black and white)
- Initial state is chosen randomly
- Different types of transition rules lead to qualitatively different global behavior



(Bedau, 2003)

Fitting Quote

“The chess-board is the world; the pieces are the phenomena of the universe; the rules of the game are what we call the laws of Nature.” (T.H.Hardy)

Components of Cellular Automata

- Cell
- State
- Lattice
- Discrete Time Development
- Neighborhood
- Boundary Conditions
- Transition Rules
- Initial Configuration

Cell

- A cell is the basic element of Cellular Automata
- A cell is a kind of memory element and stores a state

State

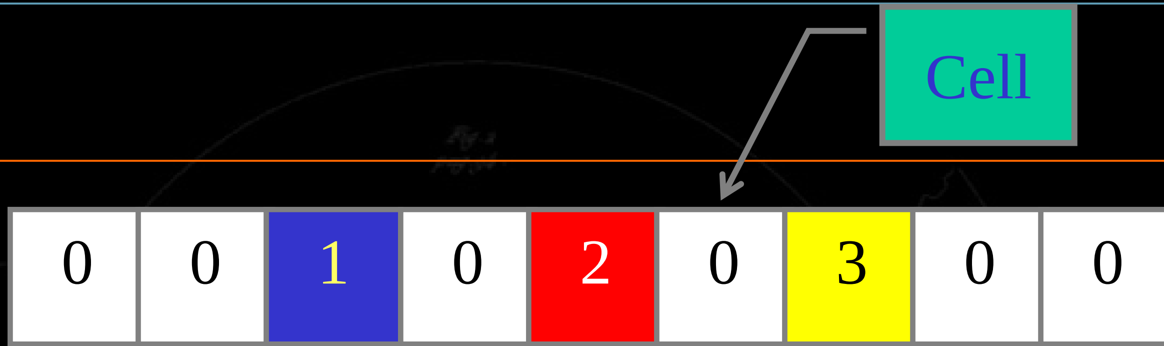
Cell i at time t is in one of a finite set Σ of k states

- often binary, $\Sigma = \{0,1\}$
- ternary, $\Sigma = \{0,1,2\}$

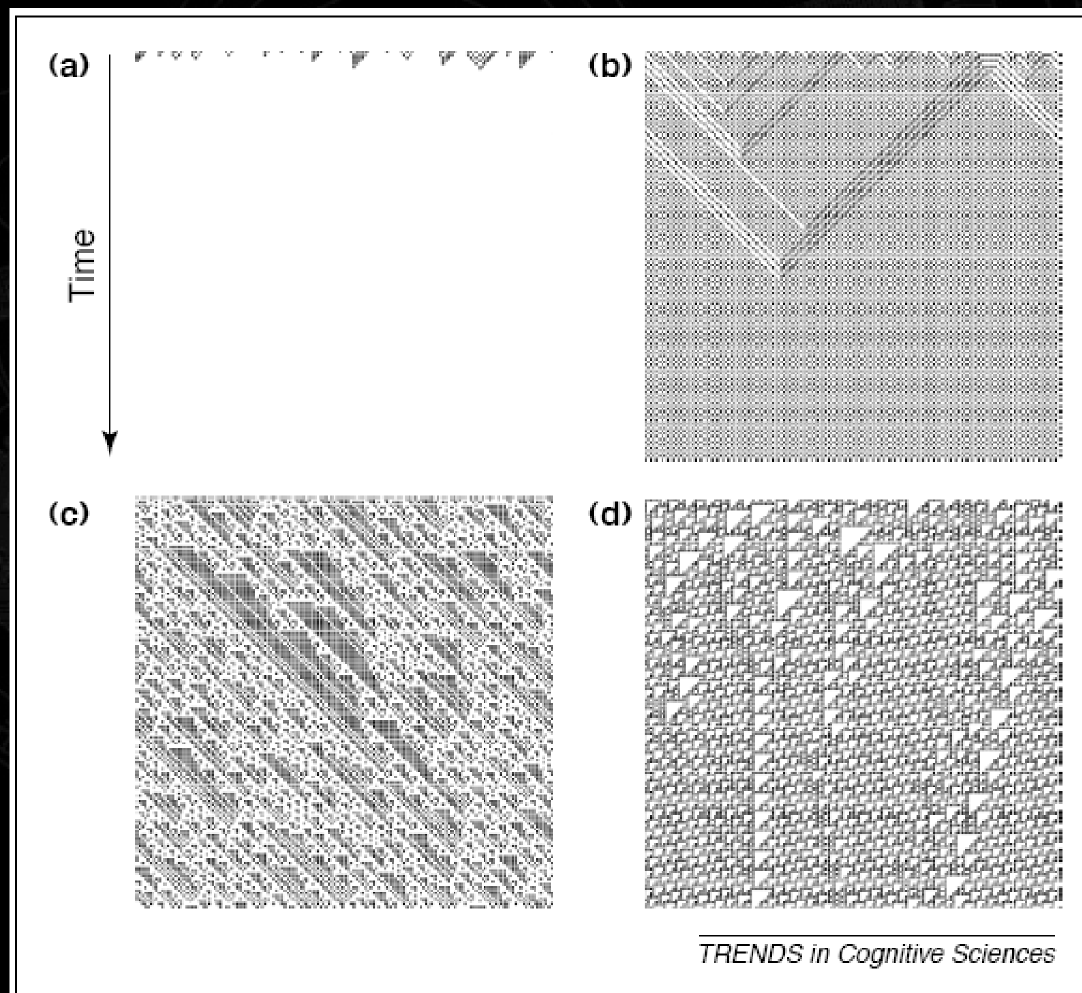
Lattice

A lattice is an (1-D, 2-D, 3-D) array in which all the cells are arranged

1-D Lattice



space-time diagram



1-D Lattice

From (Wuensche and Miller, 1992)

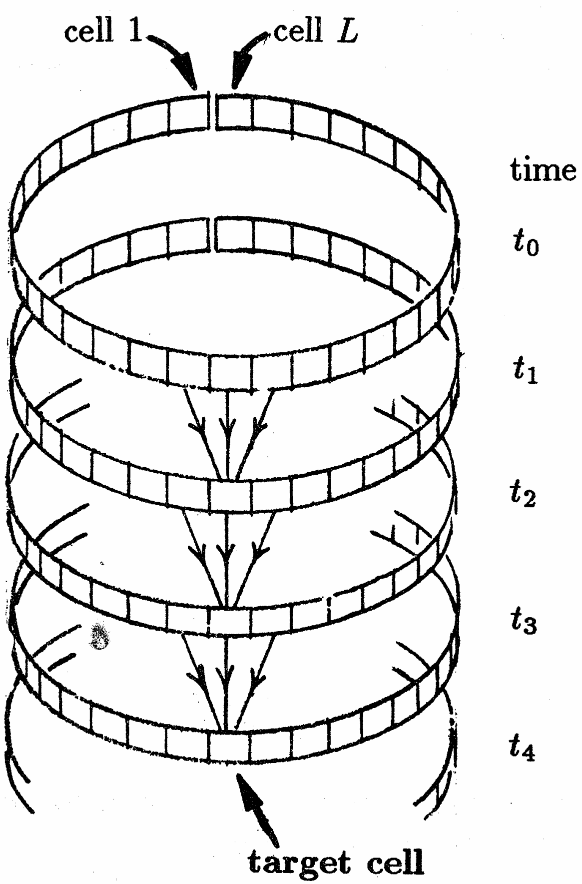
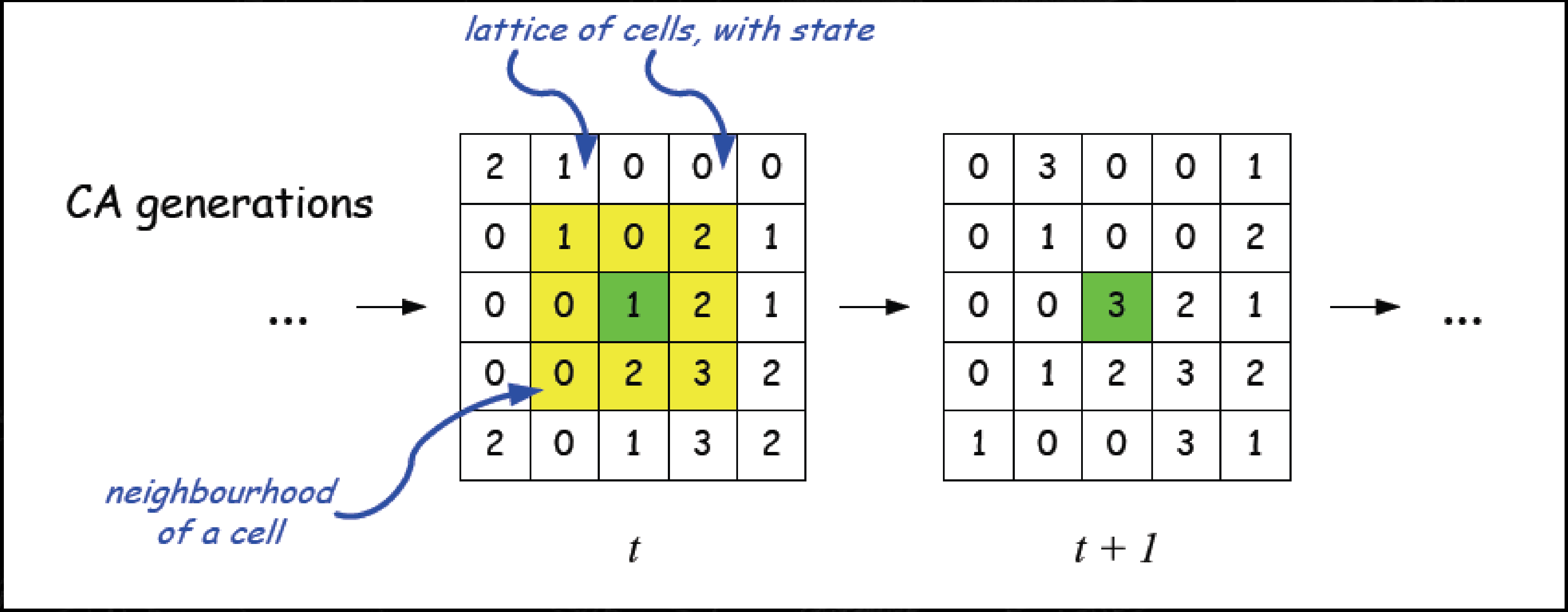


FIGURE 2.1 1-D, local binary CA with periodic boundary conditions, neighbourhood 3 (elementary rules), array length L.

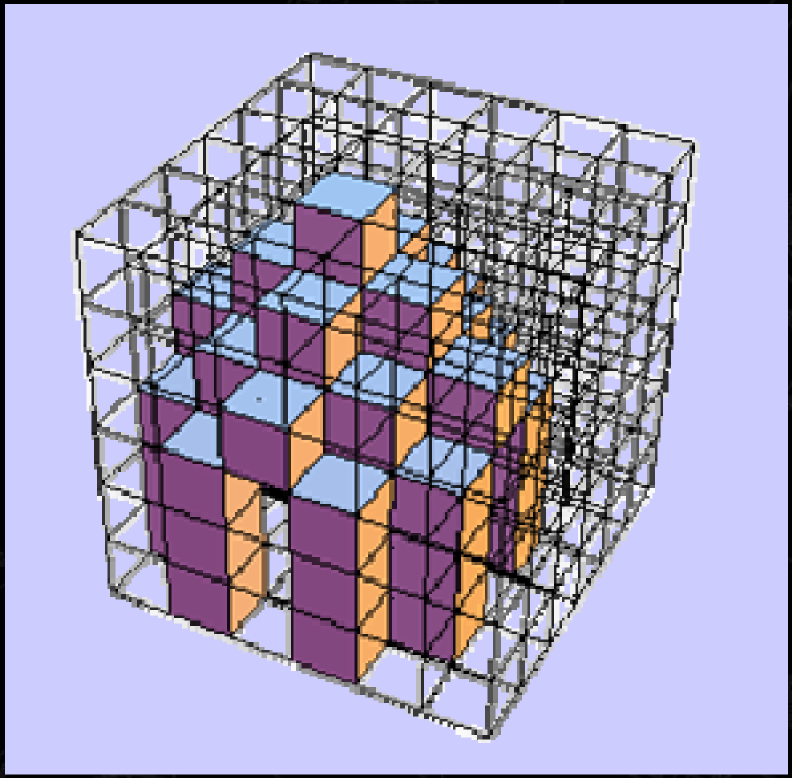
2-D Lattice

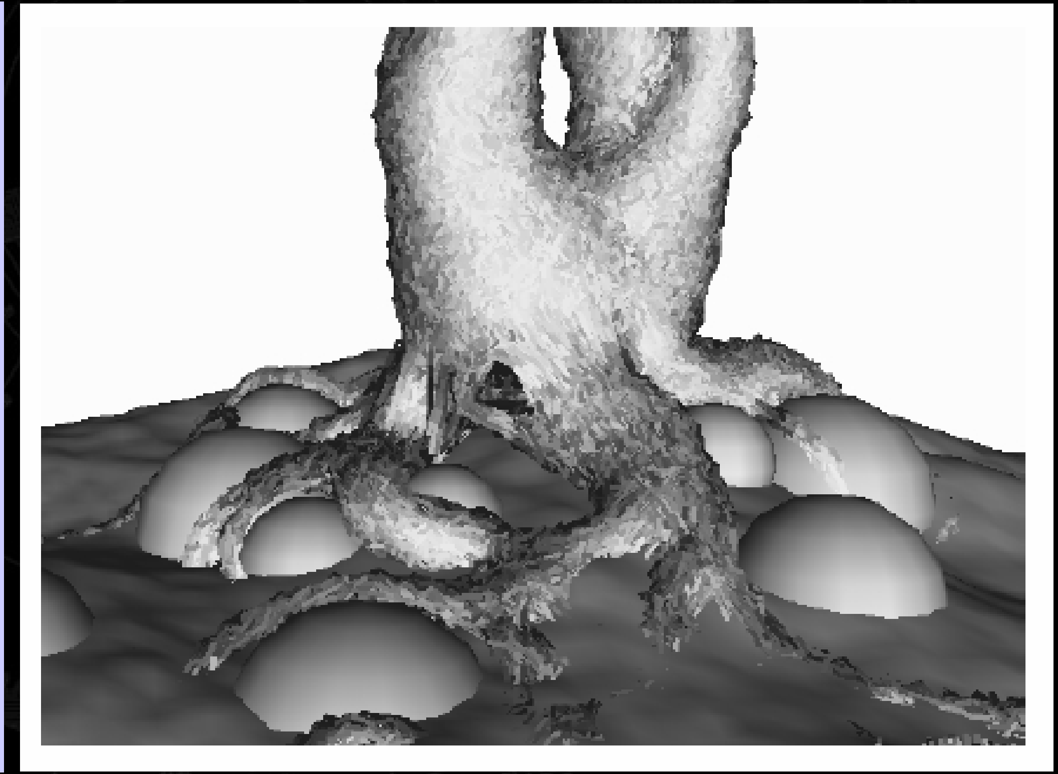


lattice of cells, with state
CA generations
...
2 1 0 0 0
0 1 0 2 1
0 0 1 2 1
0 0 2 3 2
2 0 1 3 2
0 3 0 0 1
0 1 0 0 2
0 0 3 2 1
0 1 2 3 2
1 0 0 3 1
...
neighbourhood of a cell
t
t + 1

3-D Lattice

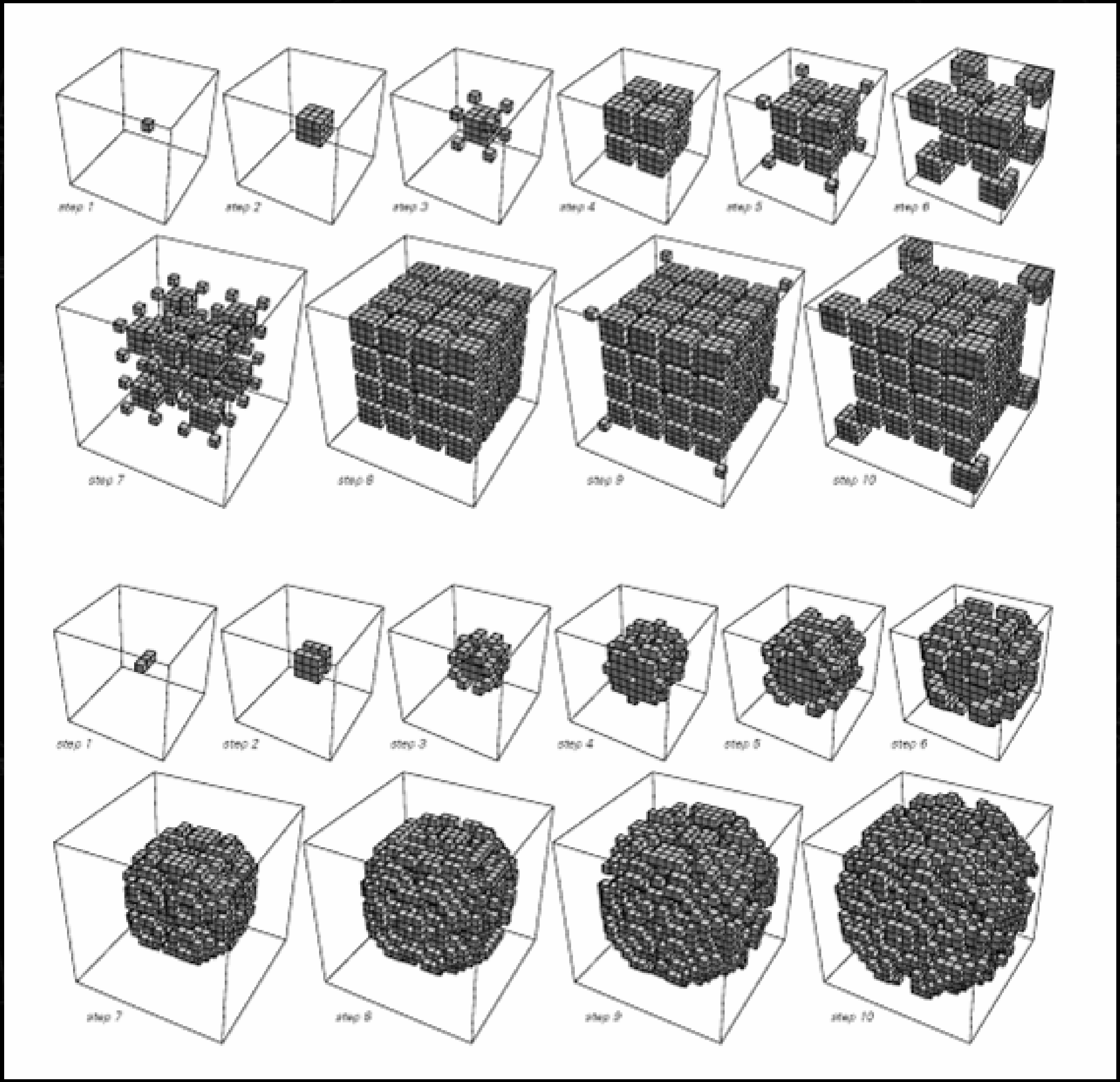
CA also called voxel automata





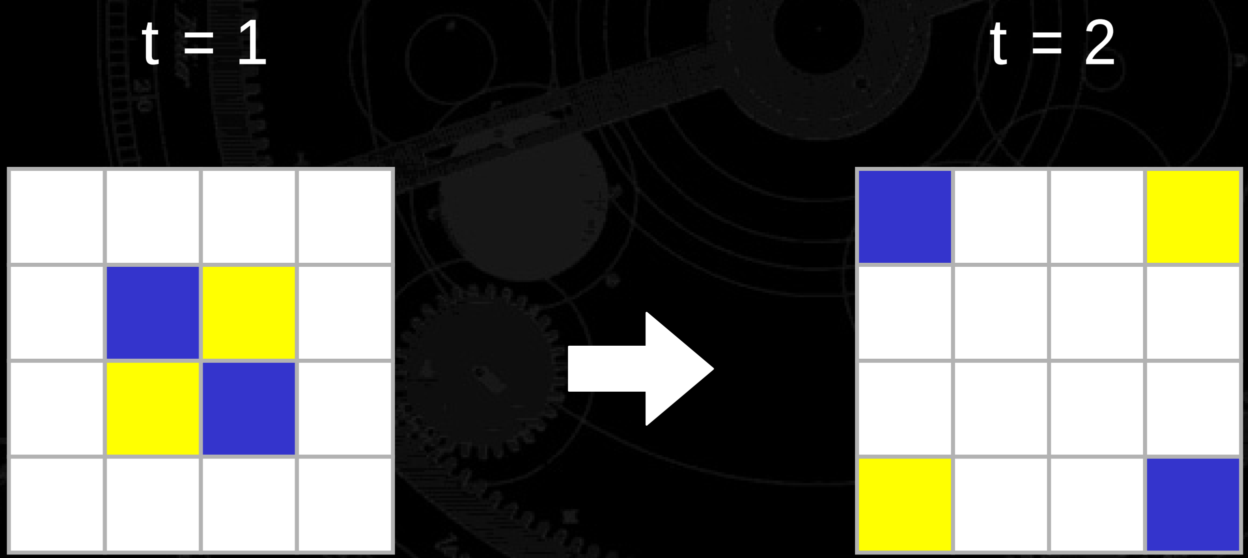
(Greene, 1989; “voxel space automata: modeling with stochastic growth processes in voxel space”)

3-D Lattice


step 1
step 2
step 3
step 4
step 5
step 6
step 7
step 8
step 9
step 10
step 1
step 2
step 3
step 4
step 5
step 6
step 7
step 8
step 9
step 10

Discrete-Time Development

States of all the cells are updated in discrete time steps

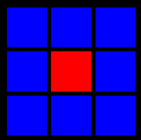


Neighborhood

- The neighborhood of a cell consists of the surrounding cells
  - N cells: itself, plus the adjoining cells it is connected to
    - For 1-D CA, $n = 2r+1$, r = "radius"
    - For 2-D CA, $n = (2r+1)^2$, r = "box range"
- Interaction is local! Meaning that no action-at-a-distance is permitted
- There can be many different definitions

Most Common Neighborhoods

von Neumann Neighborhood



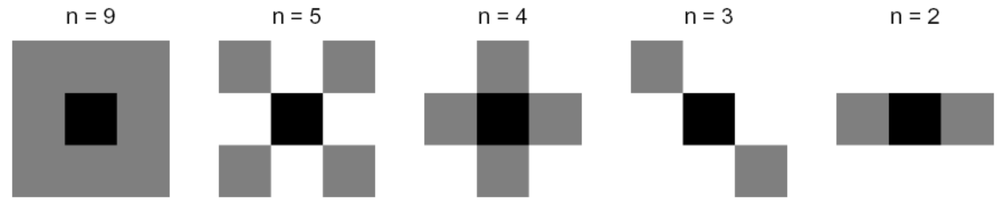
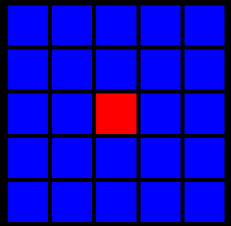
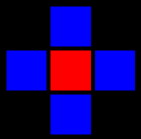


Figure 1: Depictions of Neighborhood Sizes (n). The black square is the cell the rule applies to. Gray squares are the active neighbor cells used to determine the new value.

Extended Moore Neighborhood



Moore Neighborhood



Boundary Conditions

- Infinite/adaptive grid
  - The grid grows as the pattern propagates
- Finite grid
  - Hard boundary (edge cells have a fixed state, usually zero)
  - Soft boundary (periodic boundary conditions)
    - Edge wraps around
    - 1D is a ring
    - 2D is torus
    - Weirder topologies with a twist: Moebius bands, Klein bottles

Note: a finite soft boundary CA is equivalent to an infinite grid with repeating “tiles”

Boundary Conditions

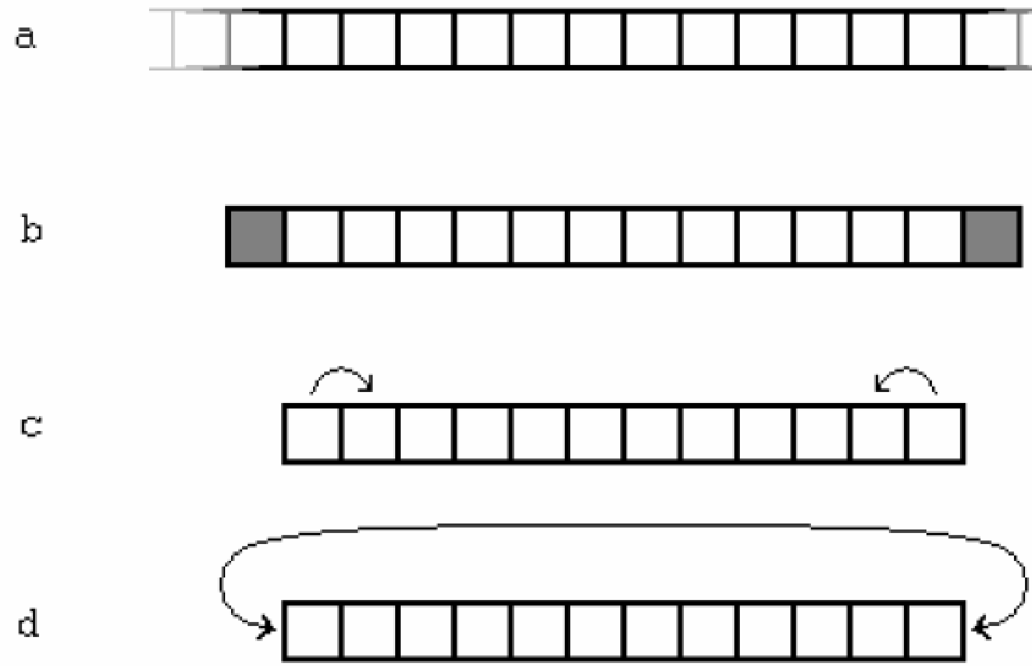


Figure 2.3: *Four possibilities for boundary conditions in a 1-dimensional CA. a) infinite (unbounded) array of cells. b) finite array of cells with fixed boundaries. The end points have cells in their neighborhood with a fixed value. c) finite array of cells with reflective boundary. The leftmost cell can only diffuse to the right. d): finite array of cells closed to a circle, periodic boundary. The leftmost cell becomes a neighbor of the rightmost cell.*

Transition Rule Table

- Transition rules determine the next states of the cells for the next time step (they define essentially a finite-state automaton)
- Depends on the lattice geometry, the neighborhood, and the state
- Typically rules are uniform, i.e. they are the same everywhere
- Different types of representations

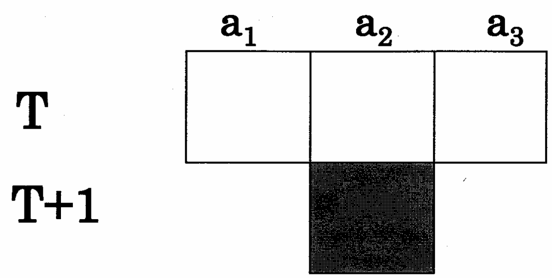
Rule Table: Representation 1

Table 2.2: Example of a simple local rule (rule table).

$a_{i-1}(t)$	$a_i(t)$	$a_{i+1}(t)$	$a_i(t+1)$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Transition Rule Table: Representation 2

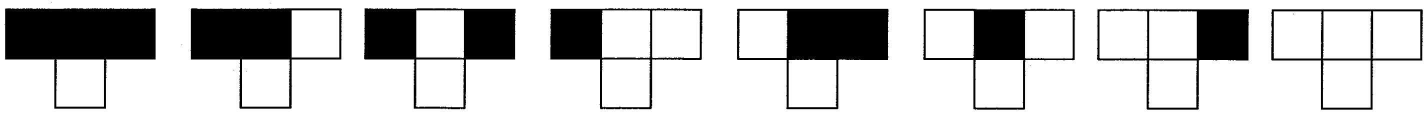
Elementary Cellular Automata



K=2

N=2

(black =1, white=0)



$$2^3 = 8$$

$$2^8 = 256$$

Showtime!

1-D CA:

http://www.artificial-life.com/en/demos/automata/automata1d.asp

3-D CA:

http://www.artificial-life.com/en/demos/geneticode/

Transition Rule Table: 1-D

Question: for a *k* state, *n* neighborhood CA, there are how many possible rules?

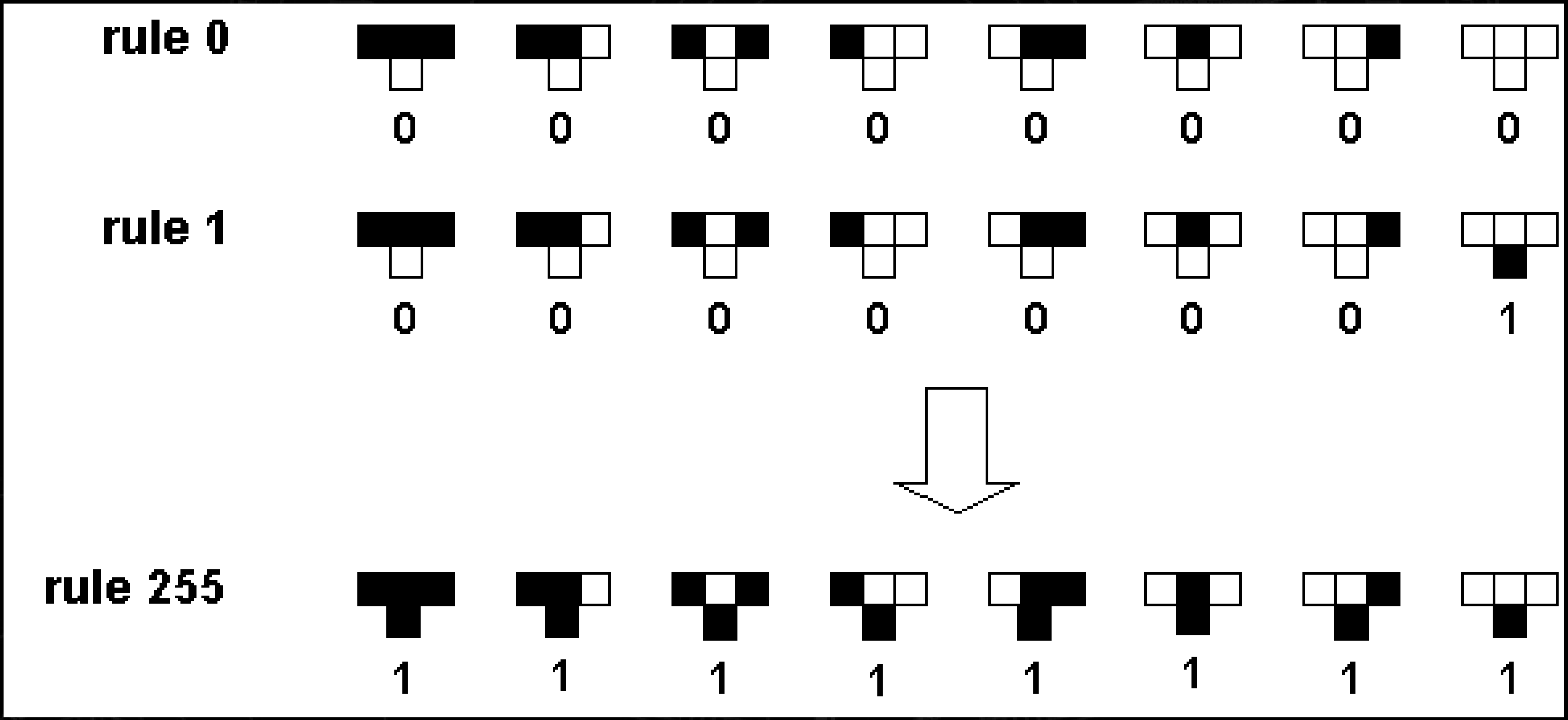
Transition Rule Table: 1-D

For k=2 and r=1 (n=3) 1-D CA there are:

- 8 possible neighbor-configuration: k^(2r+1)
- 256 different rules: k^(k^(2r+1))

→ Exhaustive enumeration is possible

Transition Rule Table: Exhaustive



Transition Rule Table: Intractable

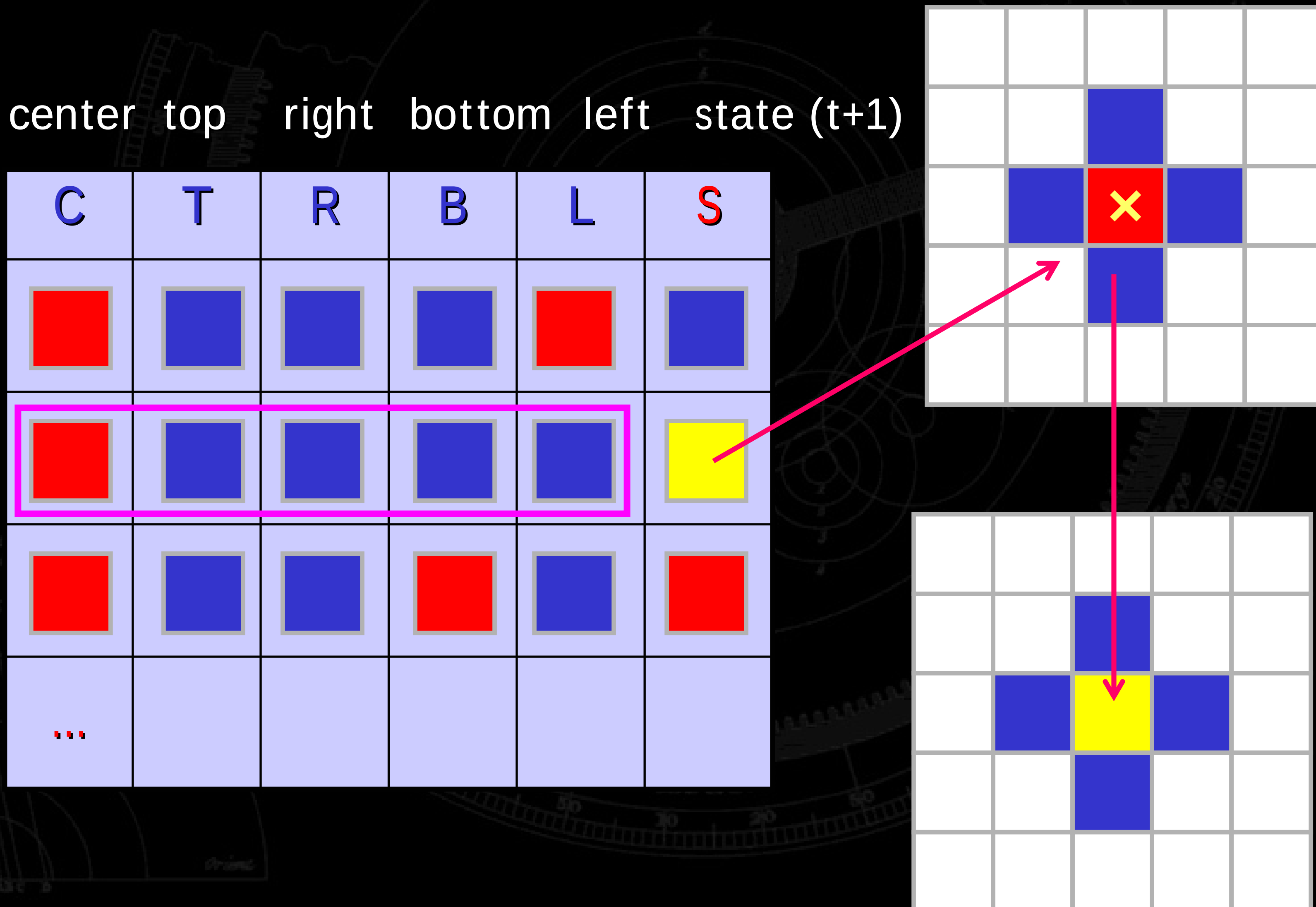
k = 10 (10 states)

r = 2 (4 neighbors)

size of rule table = k^(2r+1) = 10^5 = 100’000 rules

# of rule table = 10^100’000!!!

Transition Rule Table: 2-D



Transition Rule Table: 2-D

a bit more hard-core ...

CTRBL->I	CTRBL->I	CTRBL->I	CTRBL->I	CTRBL->I
00000->0	02527->1	11322->1	20242->2	30102->1
00001->2	10001->1	12224->4	20245->2	30122->0
00002->0	10006->1	12227->7	20252->0	30251->1
00003->0	10007->7	12243->4	20255->2	40112->0
00005->0	10011->1	12254->7	20262->2	40122->0
00006->3	10012->1	12324->4	20272->2	40125->0
00007->1	10021->1	12327->7	20312->2	40212->0
00011->2	10024->4	12425->5	20321->6	40222->1
00012->2	10027->7	12426->7	20322->6	40232->6
00013->2	10051->1	12527->5	20342->2	40252->0
00021->2	10101->1	20001->2	20422->2	40322->1
00022->0	10111->1	20002->2	20512->2	50002->2
00023->0	10124->4	20004->2	20521->2	50021->5
00026->2	10127->7	20007->1	20522->2	50022->5
00027->2	10202->6	20012->2	20552->1	50023->2
00032->0	10212->1	20015->2	20572->5	50027->2
00052->5	10221->1	20021->2	20622->2	50052->0
00062->2	10224->4	20022->2	20672->2	50202->2
00072->2	10226->3	20023->2	20712->2	50212->2
00102->2	10227->7	20024->2	20722->2	50215->2
00112->0	10232->7	20025->0	20742->2	50222->0
00202->0	10242->4	20026->2	20772->2	50224->4
00203->0	10262->6	20027->2	21122->2	50272->2
00205->0	10264->4	20032->6	21126->1	51212->2
00212->5	10267->7	20042->3	21222->2	51222->0
00222->0	10271->0	20051->7	21224->2	51242->2
00232->2	10272->7	20052->2	21226->2	51272->2
00522->2	10542->7	20057->5	21227->2	60001->1
01232->1	11112->1	20072->2	21422->2	60002->1
01242->1	11122->1	20102->2	21522->2	60212->0
01252->5	11124->4	20112->2	21622->2	61212->5
01262->1	11125->1	20122->2	21722->2	61213->1
01272->1	11126->1	20142->2	22227->2	61222->5
01275->1	11127->7	20172->2	22244->2	70007->7
01422->1	11152->2	20202->2	22246->2	70112->0
01432->1	11212->1	20203->2	22276->2	70122->0
01442->1	11222->1	20205->2	22277->2	70125->0
01472->1	11224->4	20207->3	30001->3	70212->0
01625->1	11225->1	20212->2	30002->2	70222->1
01722->1	11227->7	20215->2	30004->1	70225->1
01725->5	11232->1	20221->2	30007->6	70232->1
01752->1	11242->4	20222->2	30012->3	70252->5
01762->1	11262->1	20227->2	30042->1	70272->0
01772->1	11272->7	20232->1	30062->2	

Transition Rule Table: 2-D

Question: for a $k=2$ state, $r=1$ neighborhood CA, there are how many possible rules?

Hint: n=9=(2r+1)^2 now

Transition Rule Table: 2-D

There are 2^(2^9) = 2^512 different *k*=2, *r*=1 (*n*=9) 2-D CA

size of a rule table: k^((2r+1)^2)

number of possible rule tables: k^(k^((2r+1)^2))

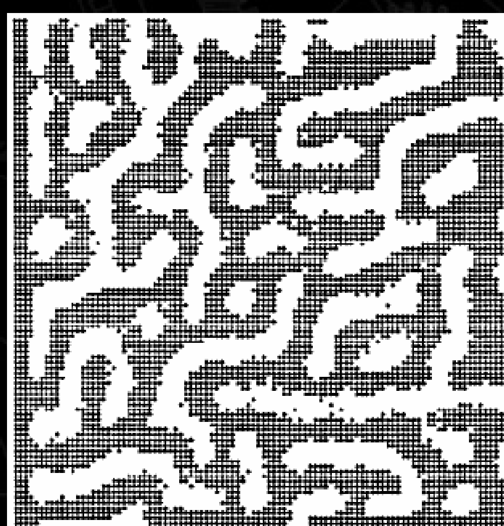
Initial Configuration

- Initial configuration defines the initial states of all the cells
- Important on development of cellular automata

Analogy to Reaction-Diffusion Models

CA can be considered a discrete-space counterpart of RD models. The space is represented by a uniform grid, with each site or cell characterized by a state chosen from a finite set

For example, Young (1984) proposed a CA model of animal coat patterns using only two states: pigmented or not



Cellular Automata: Very Brief History

1940s: CA are invented by John von Neumann by suggestion of Stanislaw Ulam (original purpose: study of process of reproduction)

1969: Konrad Zuse publishes “Rechnender Raum” (physical laws of the universe are discrete)

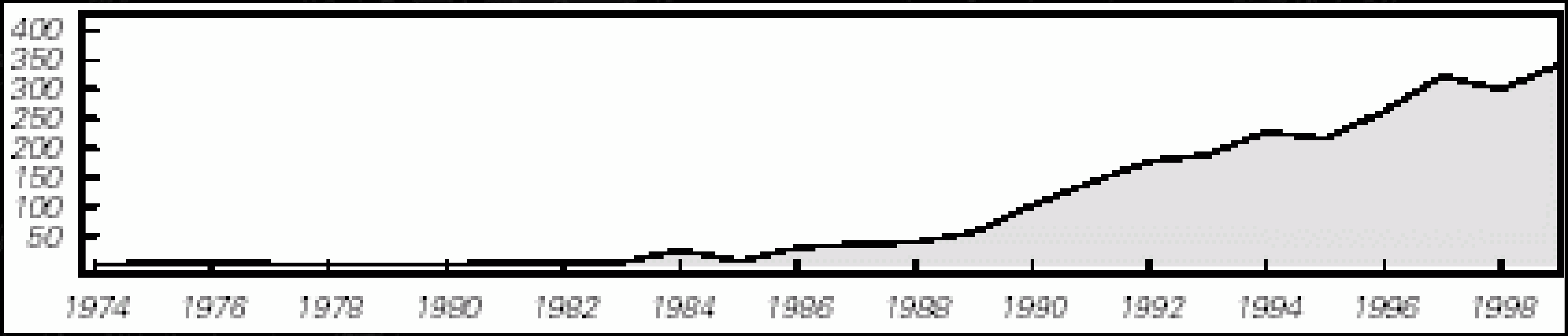
1970s: CA are popularized through John Conway’s Game of Life

1983: Stephen Wolfram publishes the first of a series of papers systematically investigating CA

2002: New Kind of Science (NKS) is published by Wolfram

Cellular Automata: Very Brief History

- Research on CA almost disappeared after the 70s (maybe because of the huge computational requirements → remember # rule tables)
- In the 80s it re-appeared with Wolfram and the number of papers published has been increasing every since (still hot!)



Cellular Automata: Very Brief History

CA have been topic of countless theses and dissertations

Concepts have been applied to numerous fields such as:

- Biological simulations
- Modeling of physical phenomena
- Computer science (e.g. computer graphics, network routing, ...)

Elementary CA

Two possible states per cell (k = 2) and a cell's neighbors defined to be the adjacent cells on either side of it (r = 1)

A cell and its two neighbors form a neighborhood of 3 cells (von Neumann neighborhood), so there are 8 possible patterns for a neighborhood (size of the rule table = 8) and therefore 256 possible rules

These 256 CAs are generally referred to using Wolfram notation. The name of a CA is the decimal number which, in binary, gives the rule table, with the eight possible neighborhoods listed in reverse counting order

Example:

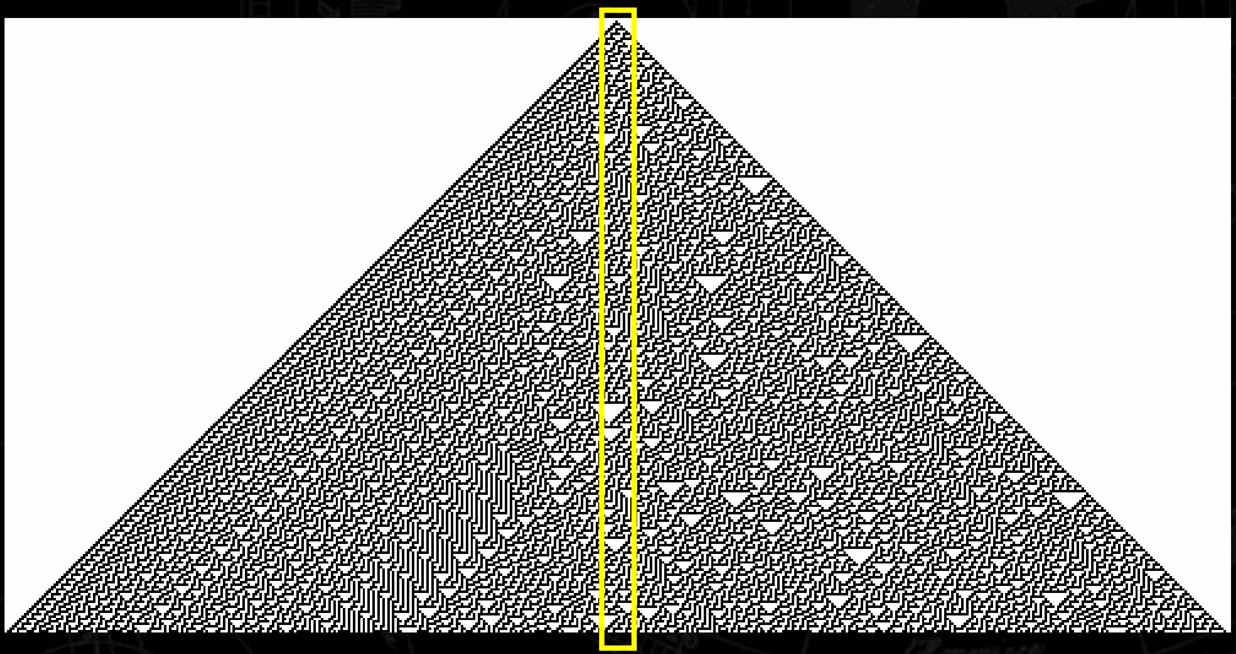
111: 0	011: 1
110: 0	010: 0
101: 1	001: 1
100: 0	000: 0

Rule 2^5 + 2^3 + 2^1 = 32 + 8 + 2 = 42 → Rule 42

Elementary CA - Rule 30

Rule 30 CA → in binary 30 = 00011110 (8 possible patterns for a neighborhood)

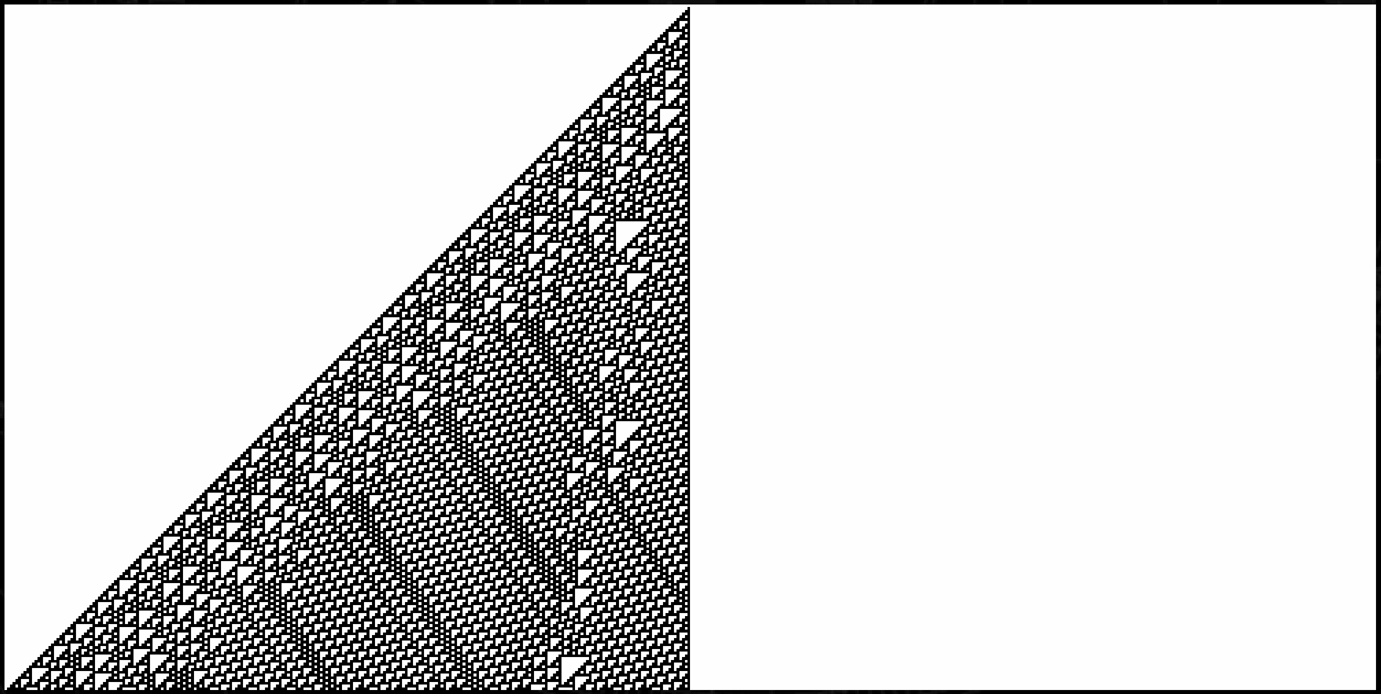
Starting from a 1 in the center



Generates randomness without random input → can be used as a PRNG!

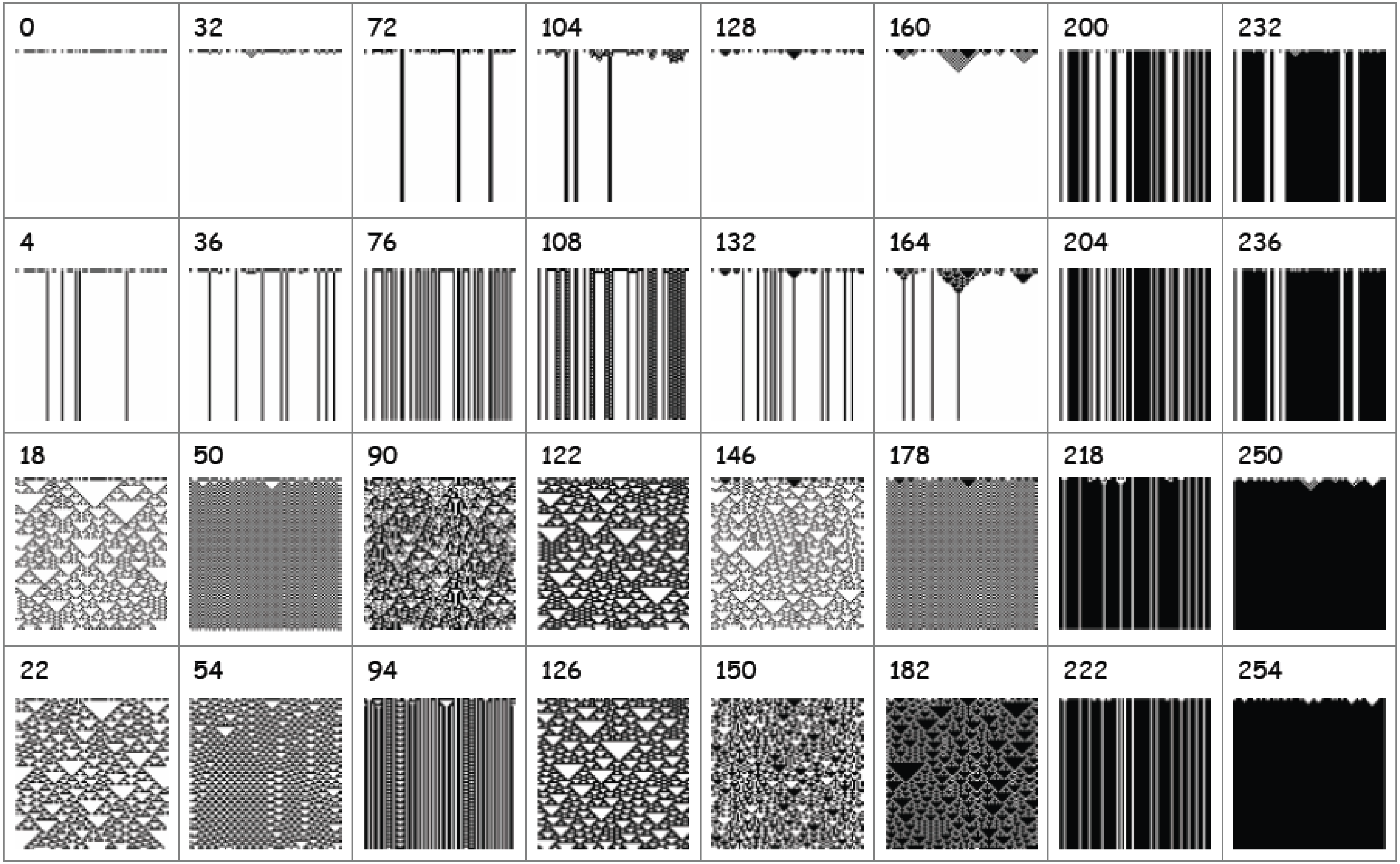
Elementary CA - Rule 110

- Rule 110 CA → in binary 110 = 01101110 (8 possible patterns for a neighborhood)
- Starting from a 1 in the center



- Neither completely random nor completely periodic → class IV CA

Space-Time Diagrams of 32 CA



Elementary Cellular Automata: Summary

small persistent patterns (~170)

- stationary
- moving

growing patterns (~85)

- repetitive patterns (~45)
- more complex patterns (36)
  - fractal, nested patterns (24)
  - random patterns (10)
  - complex mixture of regular and irregular (1)

Taxonomy of Elementary Cellular Automata

Physica 10D (1984) 1–35
North-Holland, Amsterdam

UNIVERSALITY AND COMPLEXITY IN CELLULAR AUTOMATA

Stephen WOLFRAM*
The Institute for Advanced Study, Princeton NJ 08540, USA

Cellular automata are discrete dynamical systems with simple construction but complex self-organizing behaviour. Evidence is presented that all one-dimensional cellular automata fall into four distinct universality classes. Characterizations of the structures generated in these classes are discussed. Three classes exhibit behaviour analogous to limit points, limit cycles and chaotic attractors. The fourth class is probably capable of universal computation, so that properties of its infinite time behaviour are undecidable.

Taxonomy of Elementary Cellular Automata

Class 1 – Limit points (constant)

After a finite number of time-steps, class one automata tend to achieve an *unique state* from (nearly) all possible starting conditions (complete prediction is trivial).

Class 2 – Limit cycles (repeats)

CA in this class form periodic structures that endlessly cycle through a fixed number of states (prediction requires knowledge of a finite region of sites).

Class 3 – Chaotic (pseudo-random)

From nearly all starting conditions, this type of CA's lead to aperiodic - chaotic patterns. The statistical properties of these patterns and the statistical properties of the starting patterns are almost identical (after a sufficient period of time) (prediction would require knowledge of an infinite number of sites).

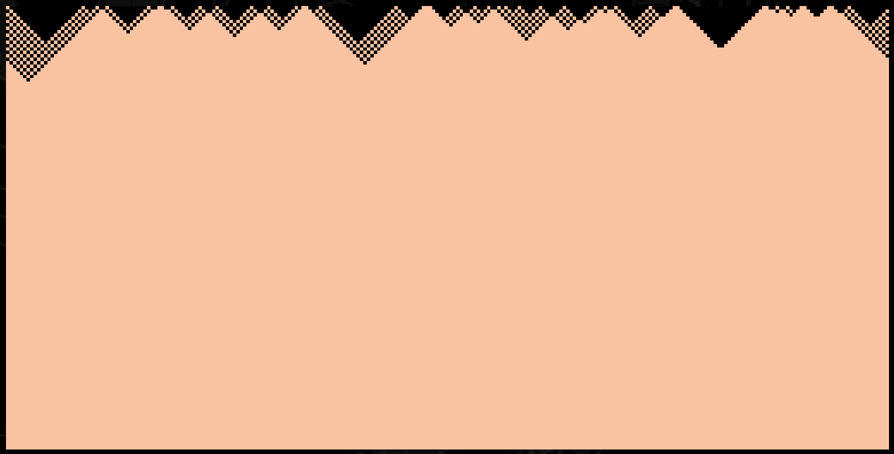
Class 4 – Structured (complex)

CA in this class form complex patterns with localized structures that move through space over time. Unpredictable and incompressible (Note: For finite CA, the patterns eventually become homogenous, as in Class 1, or periodic as in Class 2)

Class 1 – Constant

- Degenerate, single color, homogenous
- Nothing really interesting or surprising, all cells are either on/off

Rule 128, dens. = 80%

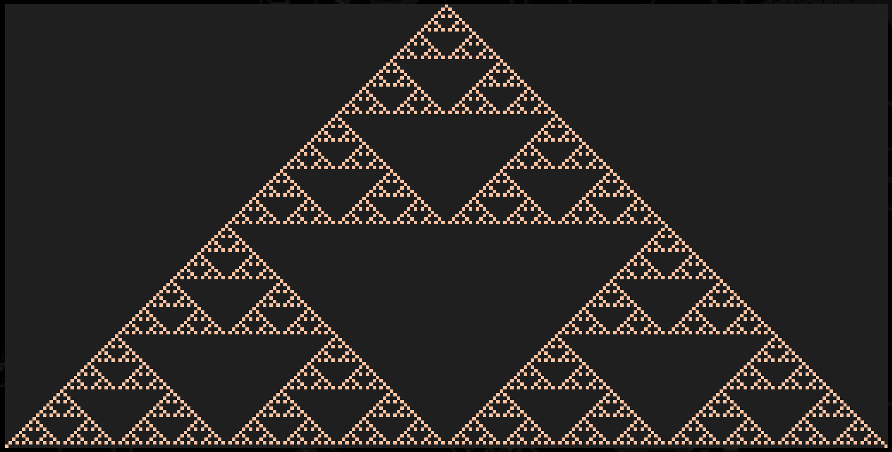


Rule 250, dens. = 20%

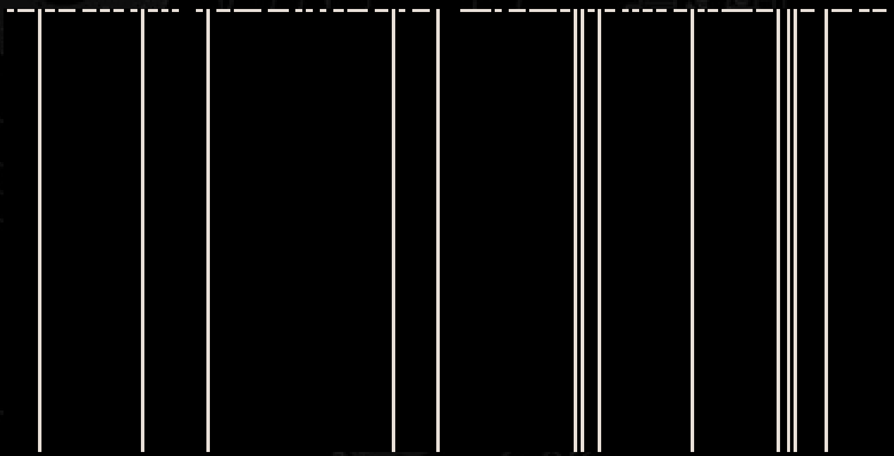
NetLogo!

Class 2 - Repeats

Stable or periodic structures, nested patterns, e.g. rules 4, 90



Rule 90

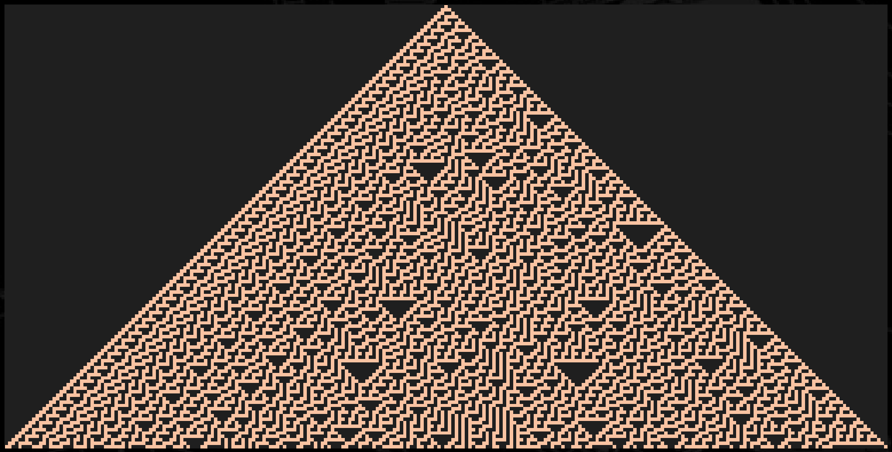


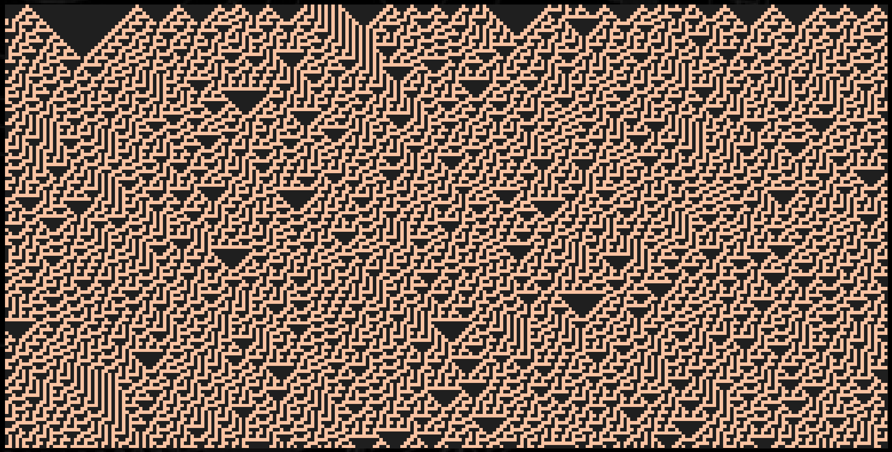
Rule 4

NetLogo!

Class 3 – Pseudo Random

- 10 rules of elementary CA are random
- Statistical analysis show randomness
- Random patterns less complex than localized structures



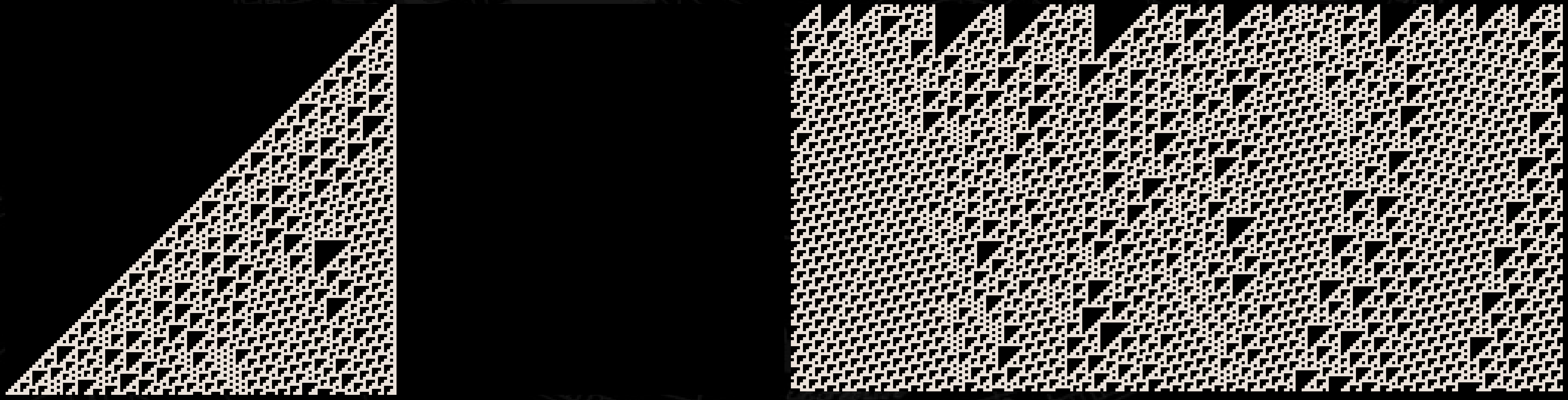


Rule 30

NetLogo!

Class 4 – Complex

- Beyond "randomness" (Wolfram)
- Neither regular nor completely random
- Unique rule among 256 elementary CA
- Capable of universal computation (example of a universal computer?)

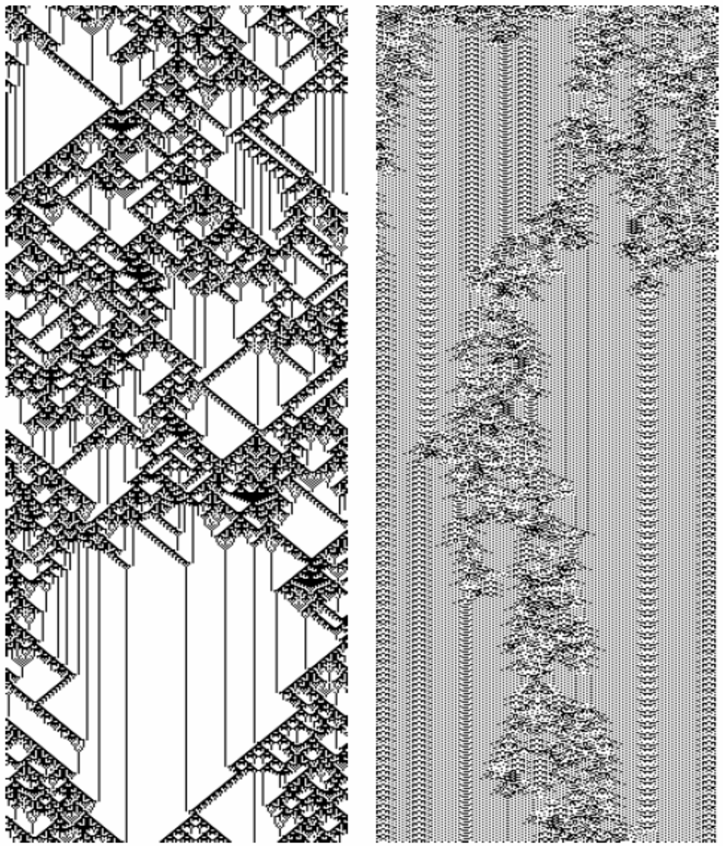


Rule 110

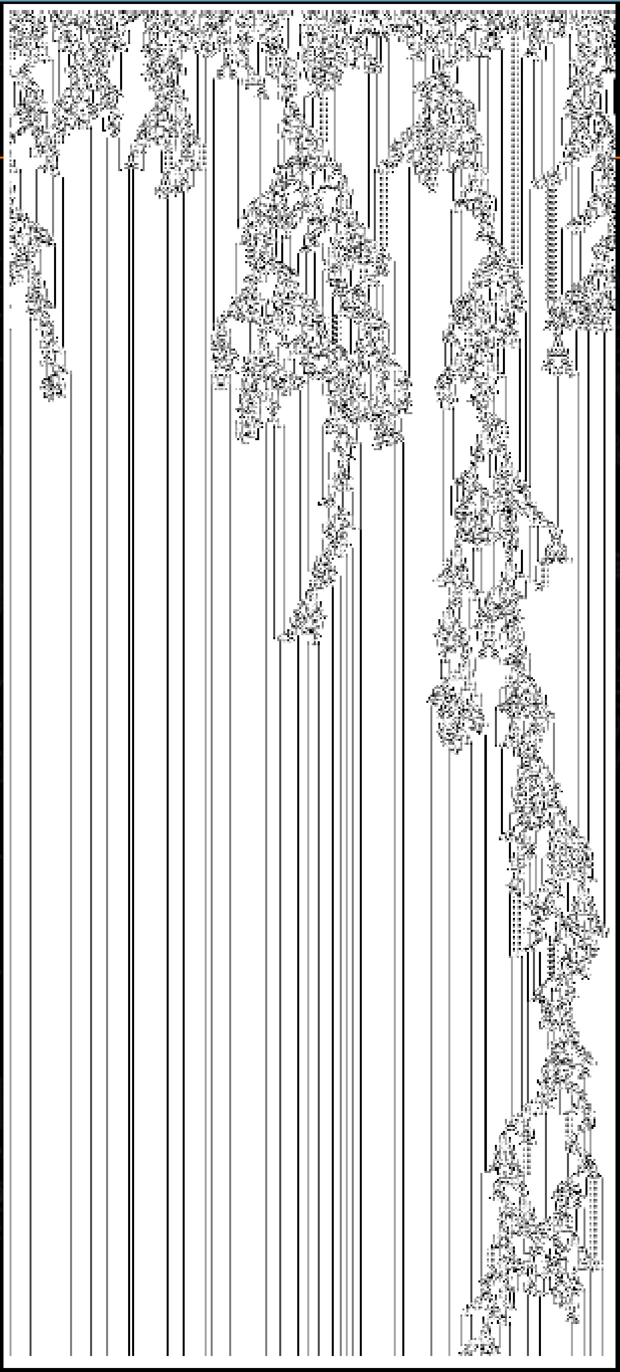
NetLogo!

Non-Elementary CA

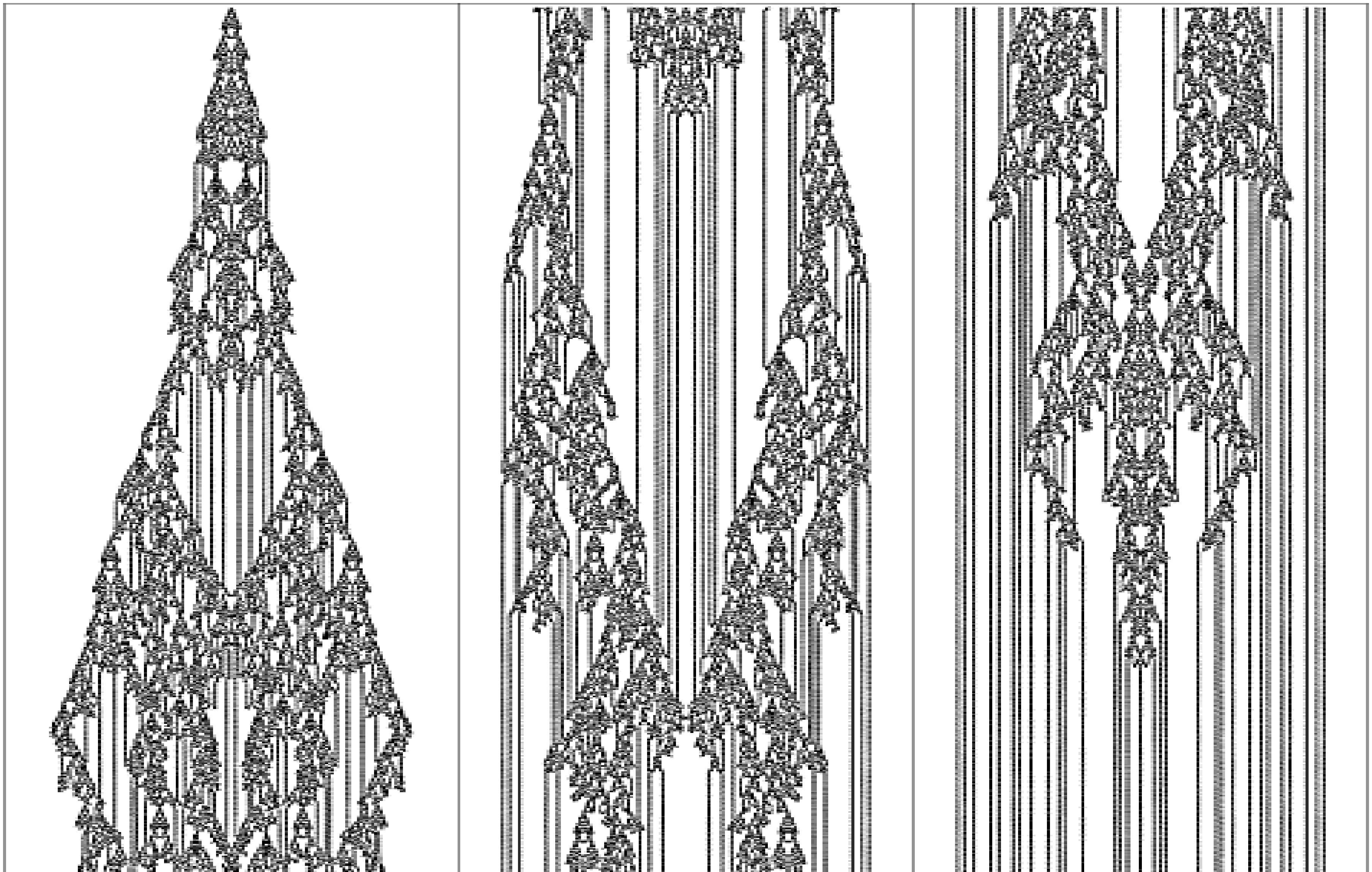
- k >= 2, r > 1
- for k > 2: colored CA



$k = 2, r = 2$ $k = 2, r = 3$

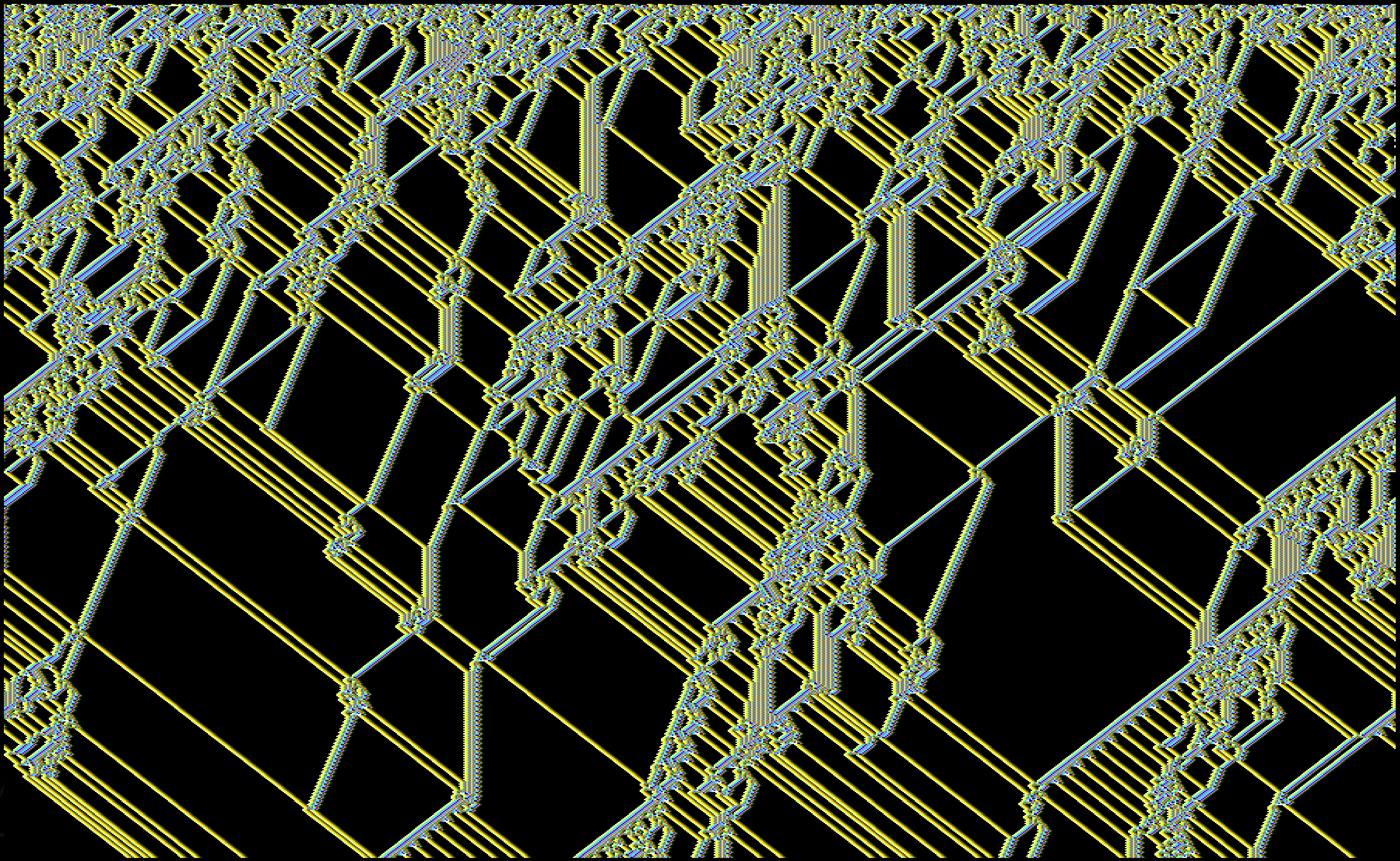


3-Color CA



code 1599

5-Color CA (n=900!) (Rule = 6c1e53a8)



Colored CA: # Rules

E.g. k = 3, r = 1

Size of rule table: 3^(2*1+1) = 27

rule tables: 3^27 = 7.626*10^12 (more than 7 trillions!!)

Totalistic Cellular Automata

A totalistic cellular automaton is a cellular automaton in which the rules depend only on the total (*sum* or *average*) of the values of the cells in a neighborhood

The evolution of a 1-D totalistic cellular automaton can be completely described by a table specifying the state a given cell at "t+1" based on the *sum/average* value of the neighboring cells and of the cell itself at time "t"

Totalistic Rules

The value of a site depends only on the sum of the values of its neighborhood, not their individual values

- For binary (k=2) rules, this reduces to the total number of "live" sites in the neighborhood
- For a k-state, n neighborhood CA, there are $k^{n(k-1)+1}$ totalistic rules
  - There are 16 different k=2, r=1 (n=3) totalistic 1-D CA
  - There are 2^10=1024 different k=2, r=1 (n=9) totalistic 2-D CA
- "outer totalistic" rules
  - Next value of a site depends on sum of the values of its outer neighborhood, and on central site (e.g. "Game of Life")

Note: totalistic rules encompass all 4 Wolfram classes of CA behaviors

3-Color Cellular Automata (non-tot. vs. tot.)

Non-totalistic rules:

More than 7 trillion rules

(Why?)

Totalistic rules:

- New color from average of previous colors
- 6561 rules

More complexity in rules

Lesson: similar behavior to elementary CA

CA Totalistic Code Numbers

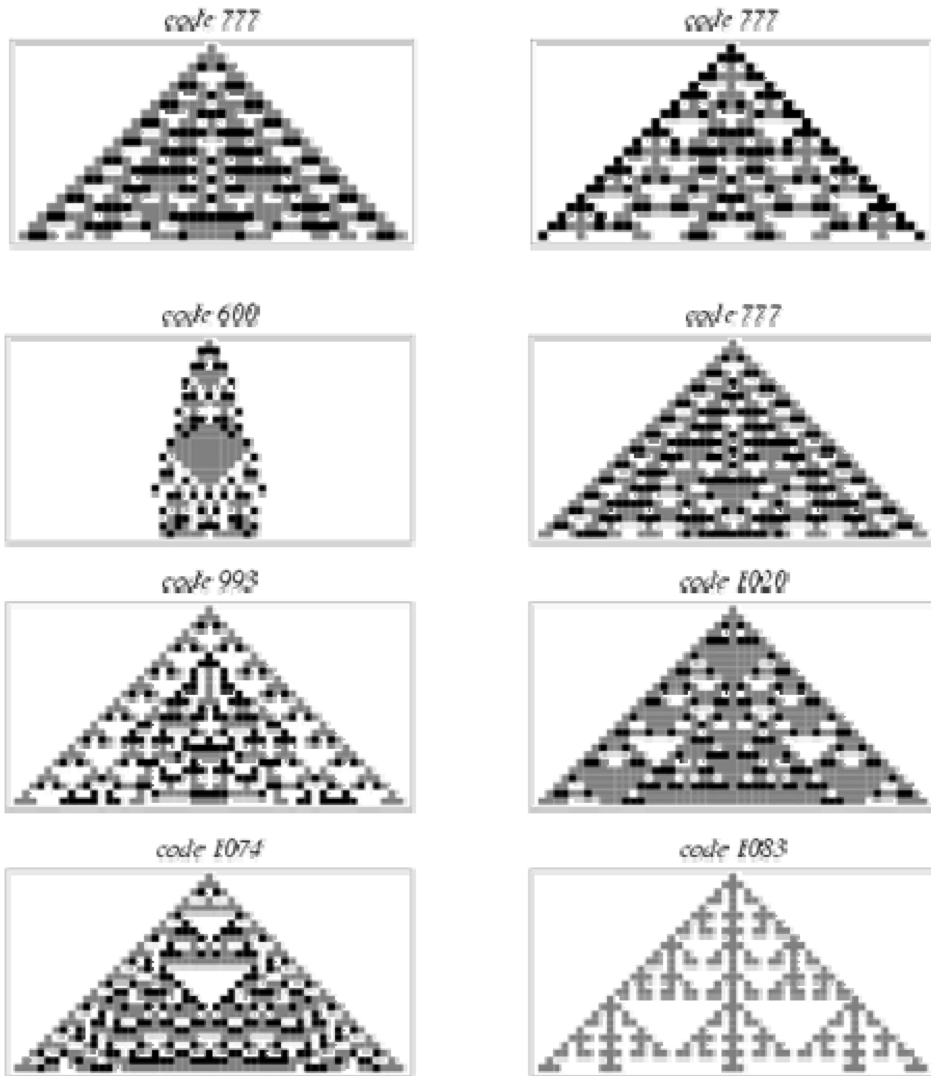
- How to refer to totalistic rules
  - Enumerate all $n(k\text{-}1)+1$ possible sums of neighborhood states, in numerical order
  - Show the result value of each state
  - Interpret the resulting string as a base *k* number
- $k = 2$, $n = 3$ example:

  3 2 1 0 (all possible sums)

  0 1 0 1

  → code: “5”

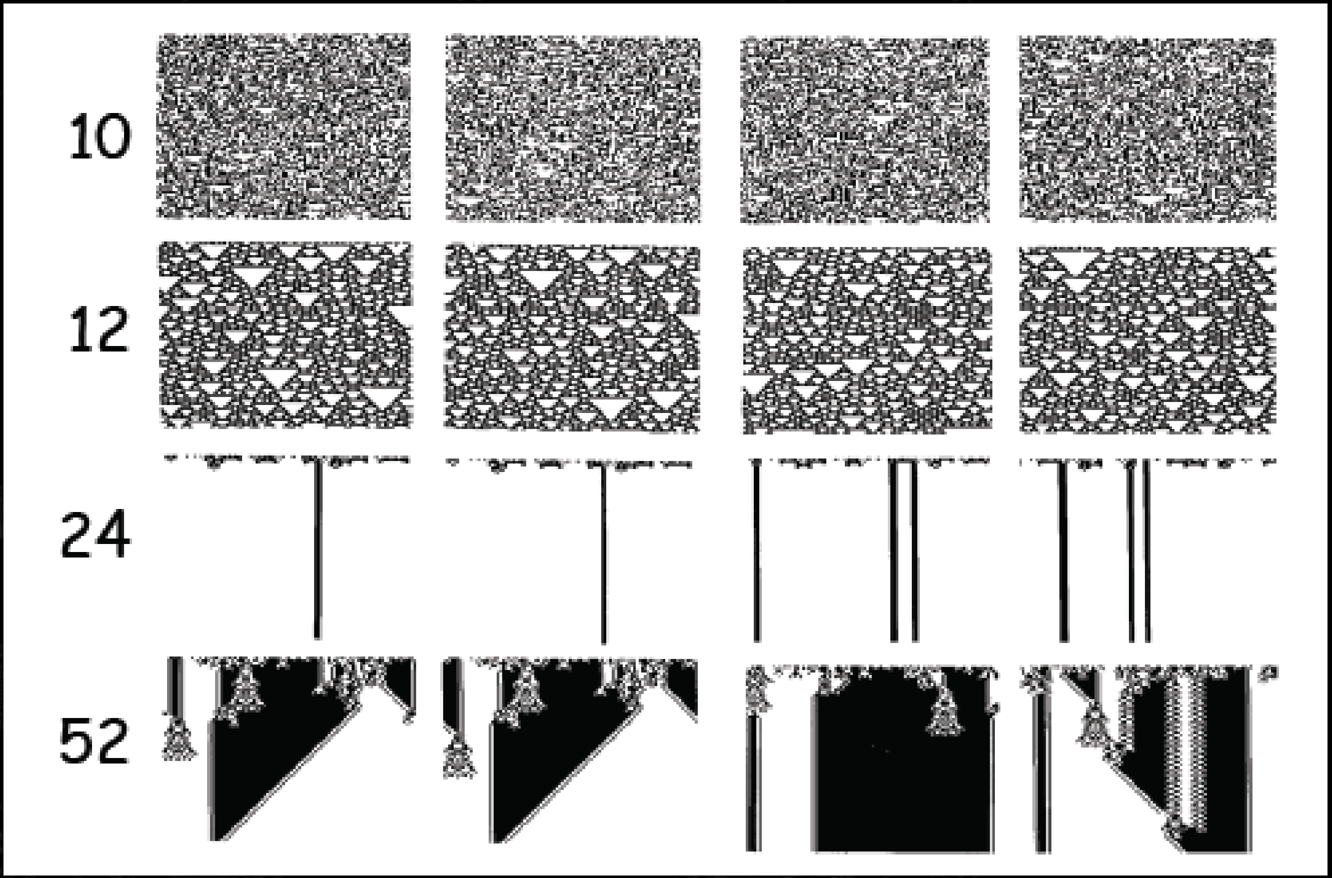
Totalistic Cellular Automata (k=3)



Dependence on Initial State

k=2, *r*=2

totalistic codes



(Wolfram, 1994)

The Lambda Parameter (λ)

- Introduced by Chris Langton in 1986 [** CLASSIC! **]
- Observation: some CA display interesting, complex, almost life-like behavior but Wolfram's classification scheme is phenomenological (argument by visual inspection of the space-time diagrams; not mathematical)
- λ is a dimensionless measure of complexity of CA; can be used to predict the behavior of a CA
- Intuitively: the probability that a neighborhood in a particular rule is mapped to an active (non-quiescent) state

The Lambda Parameter (λ)

Langton's λ derived from statistic of the rule table

Let n_0 be # of rule table entries that result in quiescent state 0, then we can define:

$\lambda = 1 - n_0 / k^n$

where k=#states, n=#neighbors (k^n=total number of rule table entries)

Example

What happens for: $\lambda = 0$, $\lambda = 1$, $\lambda = 1 - 1/k$?

The Lambda Parameter (λ)

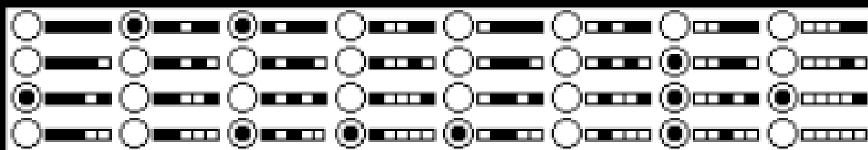
$\lambda = 0$: all transitions are to quiescent state

$\lambda = 1$: no transition to quiescent state

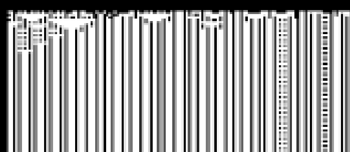
$\lambda = 1\text{-}1/k$: statistically speaking all states are represented equally in the rule table (a fraction of 1/k maps to 0)

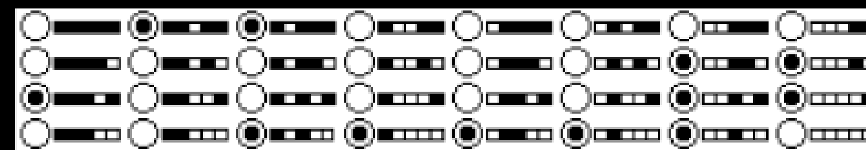
For $k = 2$, $\lambda = 0.5$, i.e. half transitions are quiescent, half are not

The Lambda Parameter (λ)

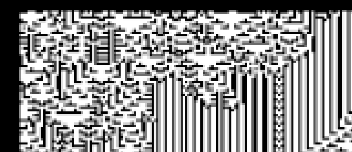


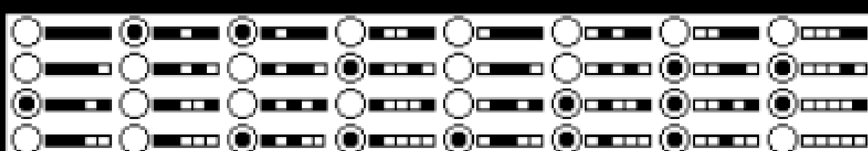
λ=10/32, Type II



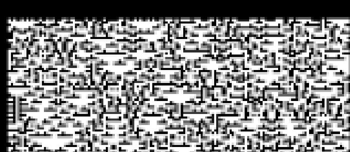


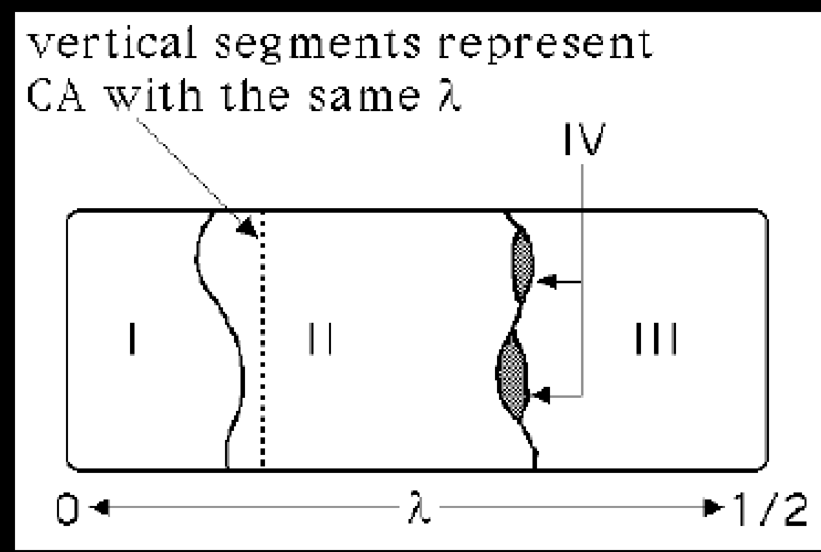
λ=12/32, Type IV





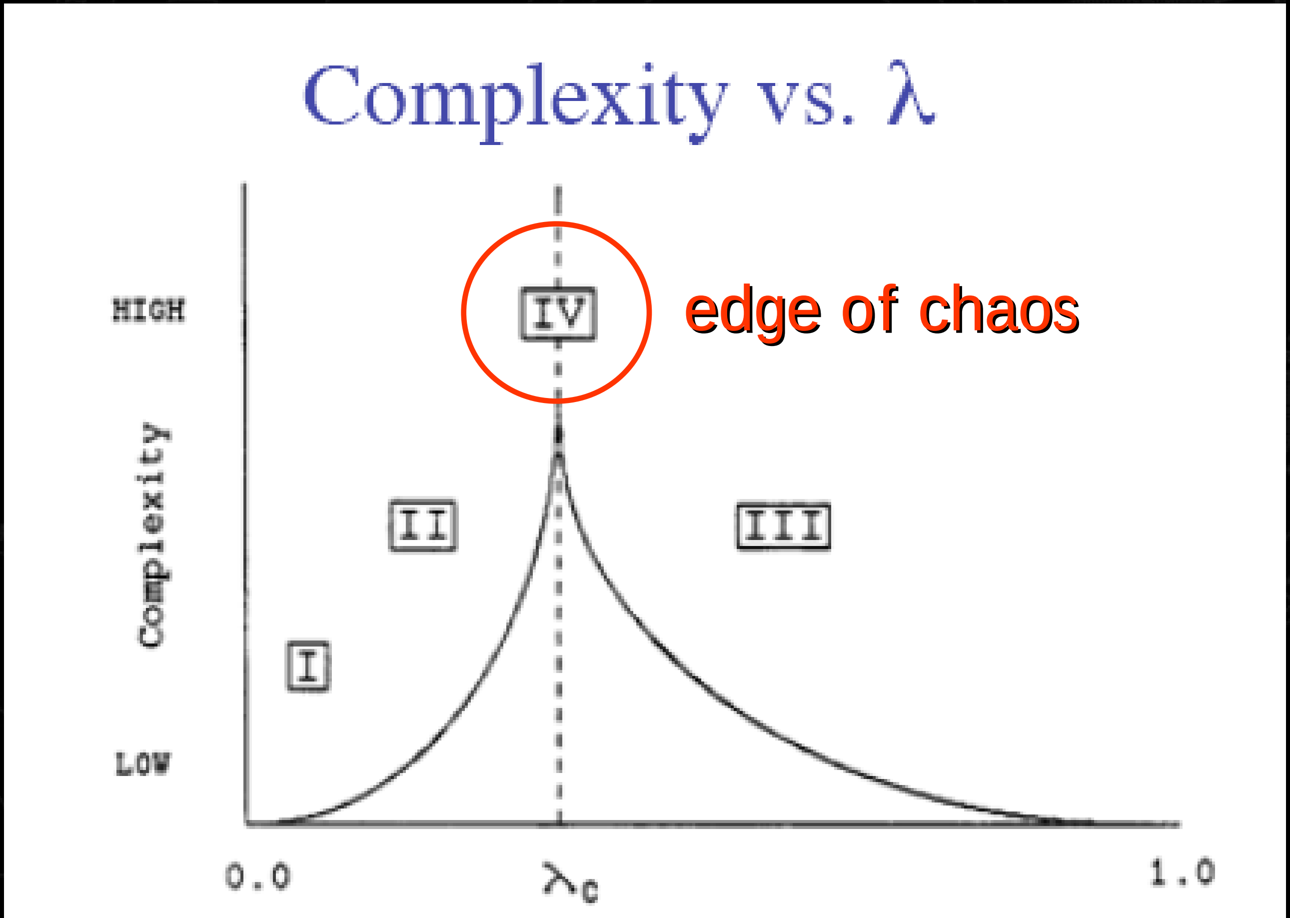
λ=14/32, Type III





The Lambda Parameter (λ)

Here, complexity = length of the transient



Is λ A Good Measure?

Rather good correlation between λ and other interesting properties for extreme values of λ, near 0 and 1

A) However, exception do exist: high λ-value can be the result of simple behavior

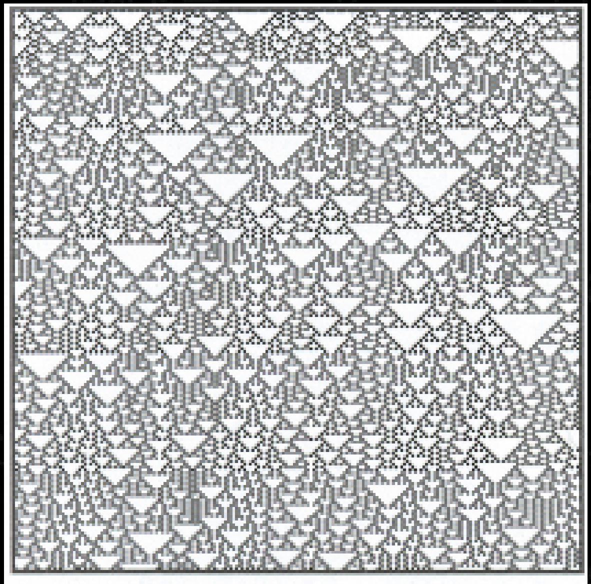
B) Performance is not so good for intermediate values, e.g. near the claimed interesting complex class IV behaviors

C) Results based on "average" behavior, but there is no "average" behavior for intermediate λ → there is great deal of variation

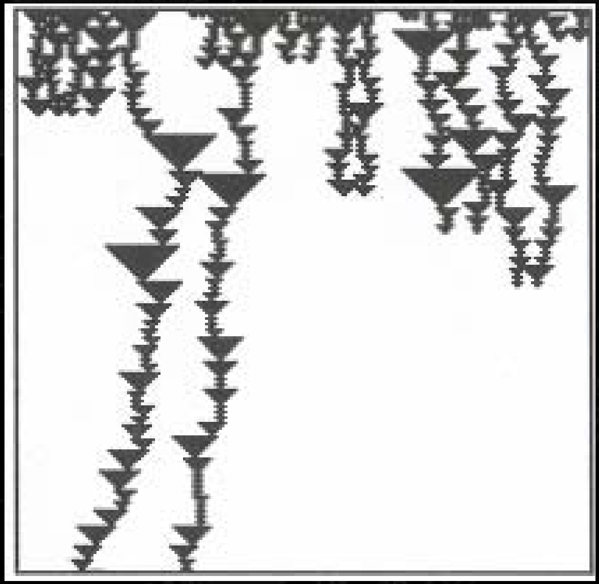
D) Given that CA rule set is the "program" and the initial configuration is the "input", CA can perform universal computation (UTM). λ parameter cannot be used to predict the long-term behavior of the CA (given a particular initial configuration) because this would be equivalent to distinguish between CA that halt ($\lambda \rightarrow 0$) and never halt ($\lambda <> 0$) → equivalent to solving the Halting Problem

More Realistic Measures

- Analysis of embedded coherent structures (Mitchell, 1998)
- Example: rule 18



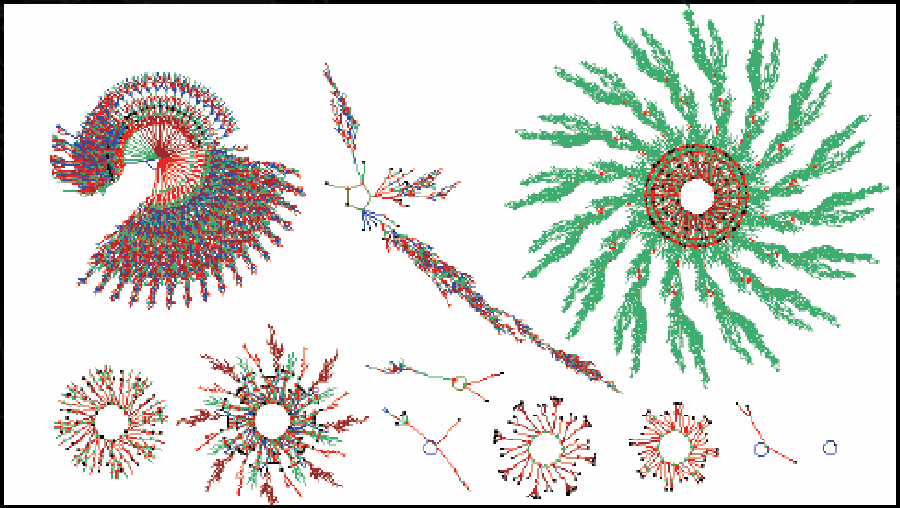
Classical space-time diagram, apparently chaotic, class III



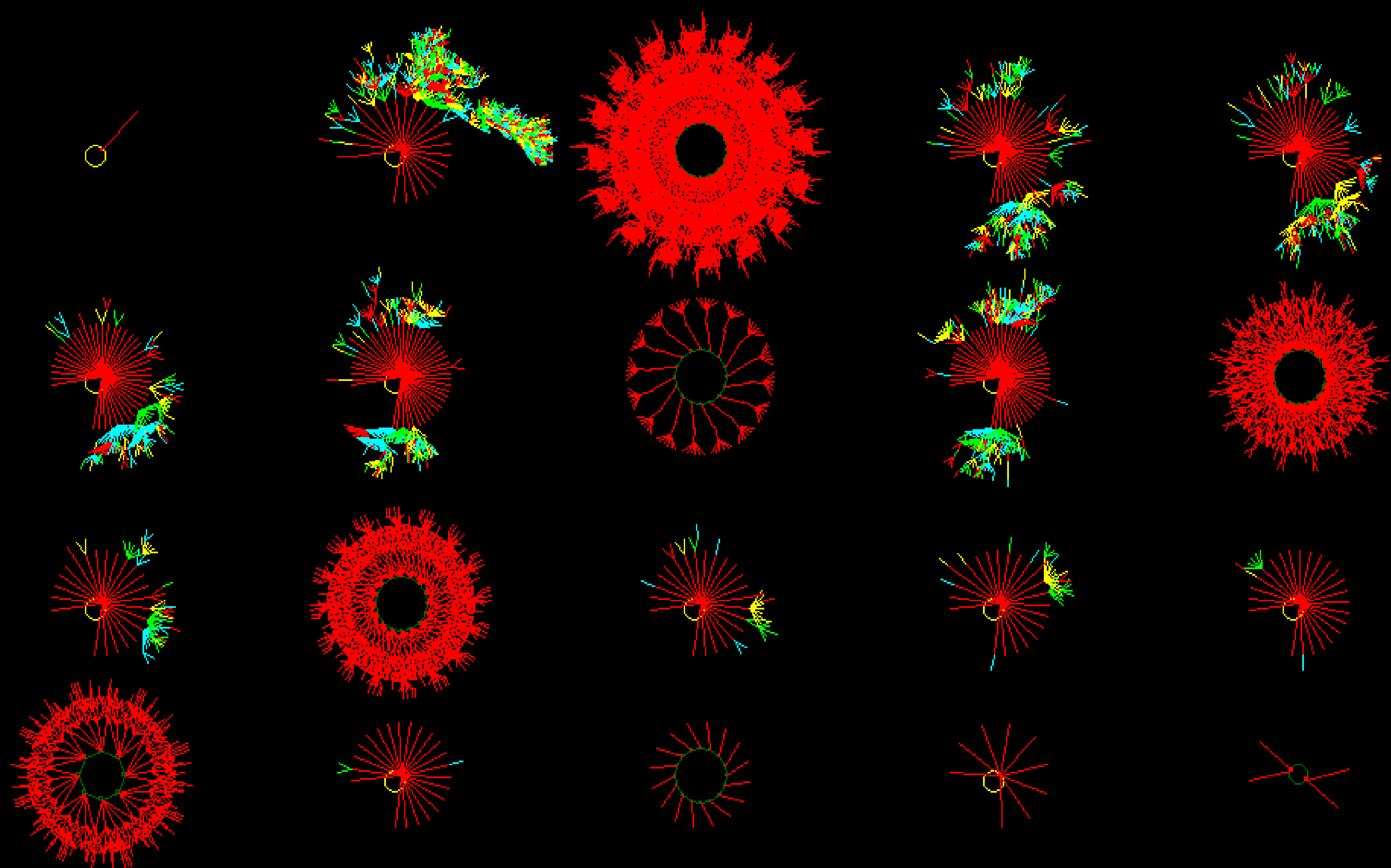
Same rule, with diagram filtered to expose underlying structure

More Realistic Measures

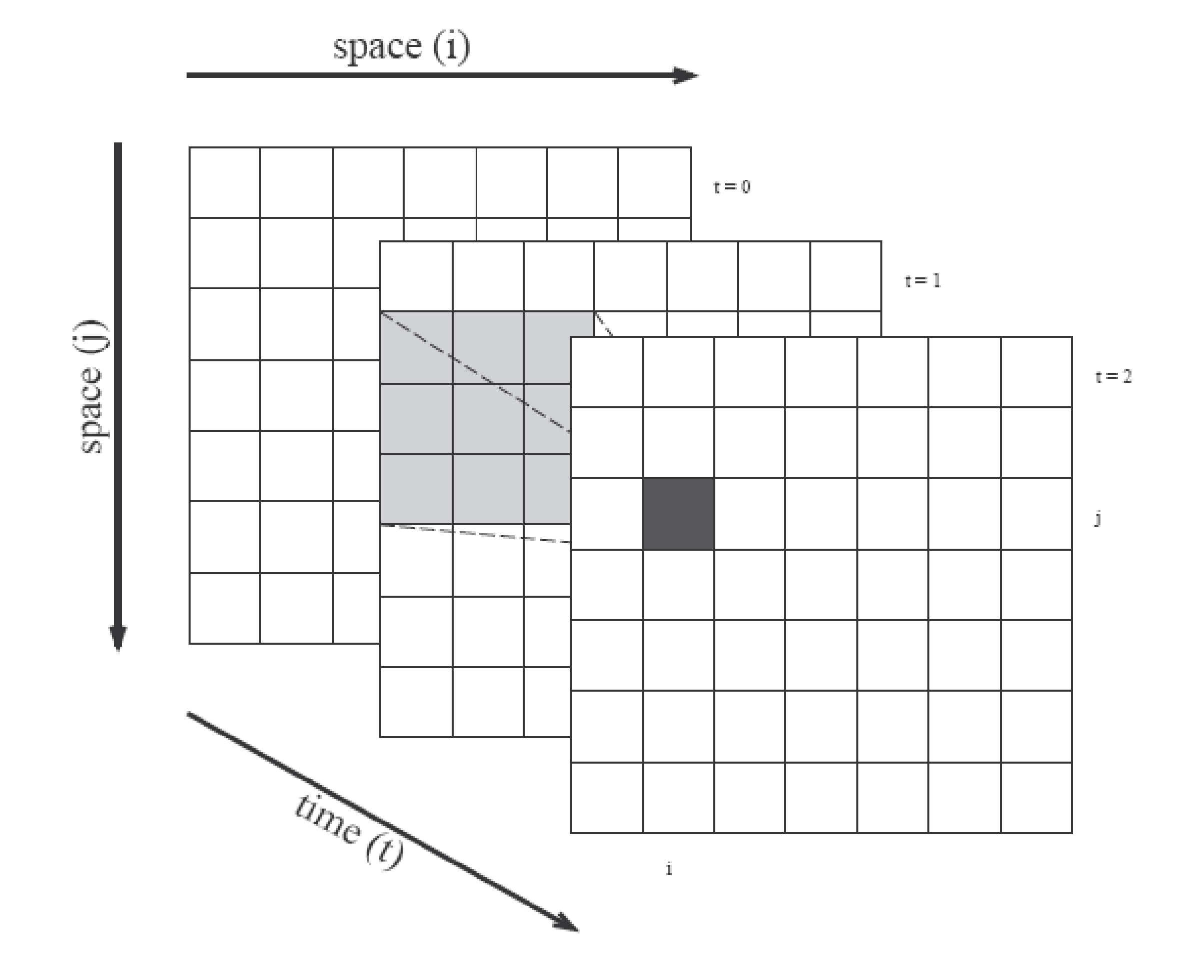
- Identify basins of attraction
- CA rules define a trajectory through its space
- Eventually a trajectory encounters a previously-seen state (the state space is finite), and an attractor cycle is formed (the trajectory leading to the attractor is a transient)
- Attractor + transitive closure = basin of attraction



More Basins of Attraction (n=16)

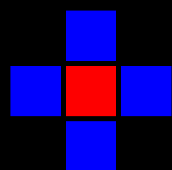


2D Cellular Automata

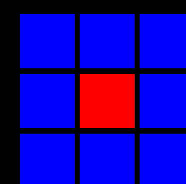


Most Common Neighborhoods

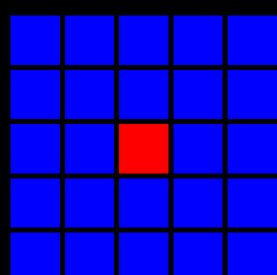
von Neumann Neighborhood



Moore Neighborhood



Extended Moore Neighborhood



Conway's "Game of Life"

The Game of Life is a CA (of class IV) devised by the British mathematician John Horton Conway in 1970 and popularized by Martin Gardner.

Original "game" was played with pieces on a Go board

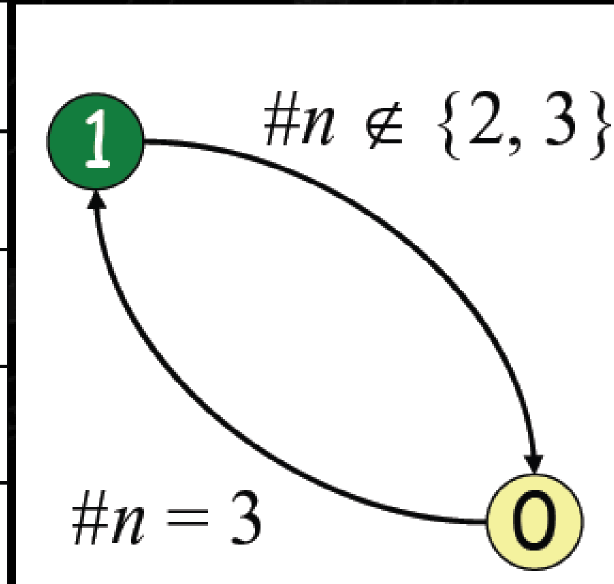
Game of Life

Essentially, a 2-D, k = 2, r = 1 (n = 9) CA

Rules of Life

- Next state depends on: sum of the 8 neighbor cell states, and on state of central cell (“outer totalistic”)

$\#n \setminus s_{i,t}$	0	1	
<2	0	0	dies of “loneliness”
2	0	1	“survives”
3	1	1	“born” / “survives”
>3	0	0	dies of “overcrowding”

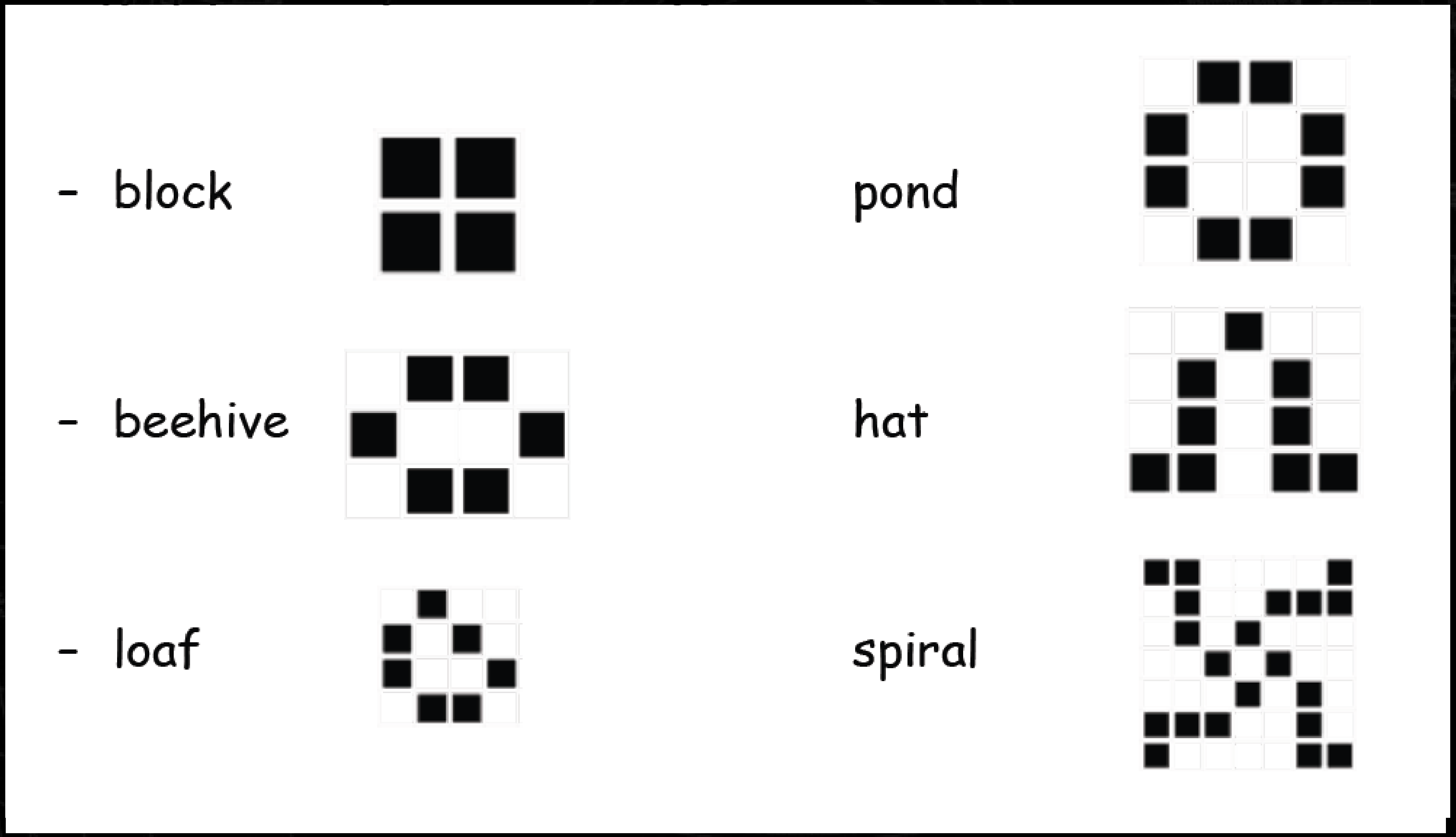


Game of Life: NetLogo!

NetLogo! + additional SW

“Still Life”

stable, unchanging pattern

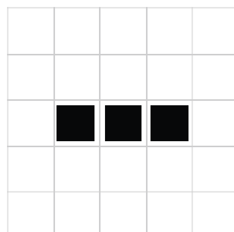


http://www.ericweisstein.com/encyclopedias/life/topics/StillLifes.html

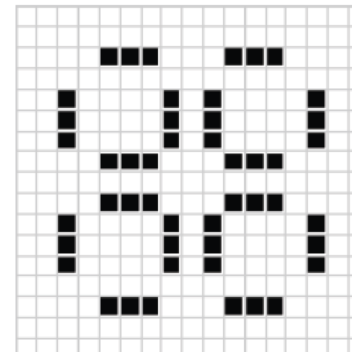
Oscillators

repeating pattern

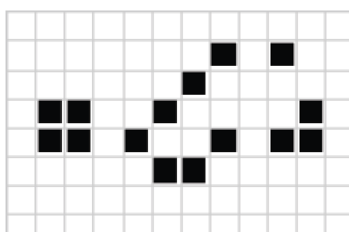
- blinker, p=2



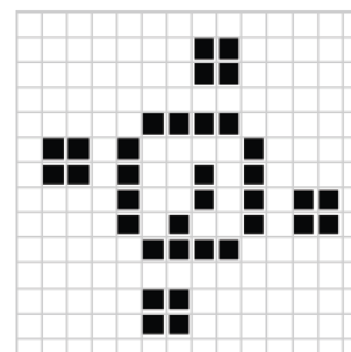
pulsar, p=3



- blocker, p=8



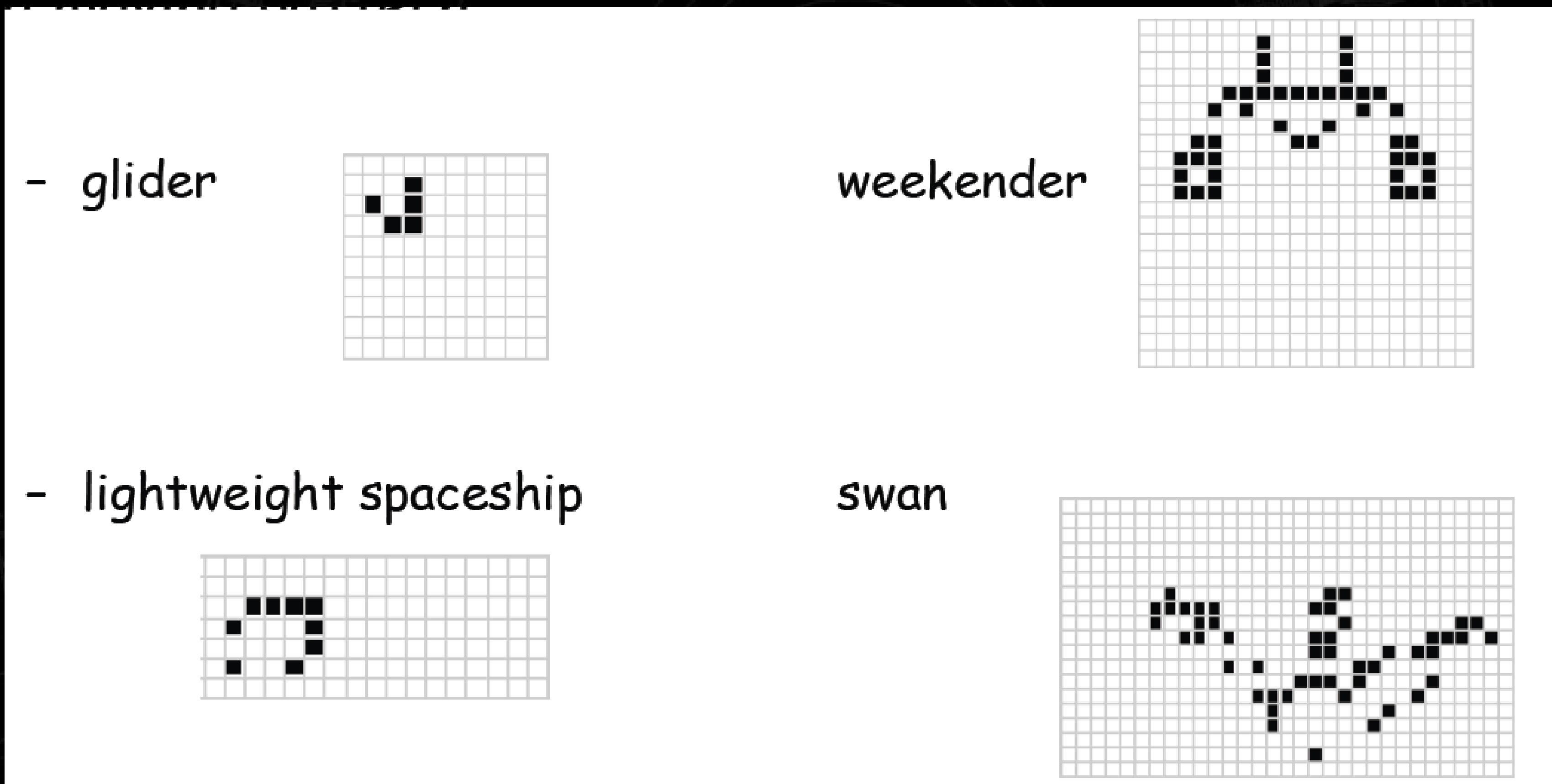
clock II, p=4



http://www.ericweisstein.com/encyclopedias/life/animations.html#movies

Spaceships

moving pattern

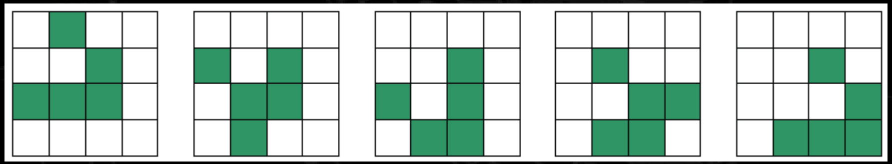


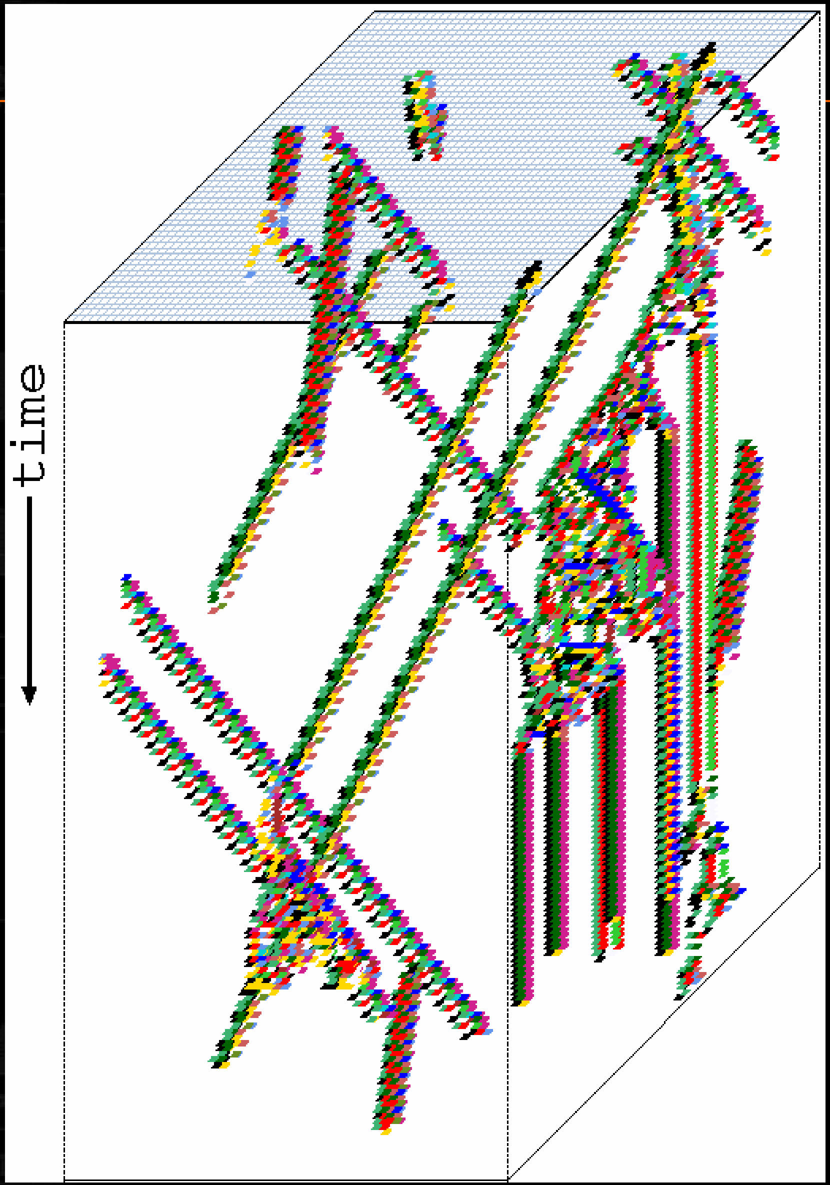
http://www.ericweisstein.com/encyclopedias/life/animations.html#movies

Glider

Smallest pattern that moves constantly across the grid

- After 4 generations the original pattern is reproduced one cell further right and one cell further down



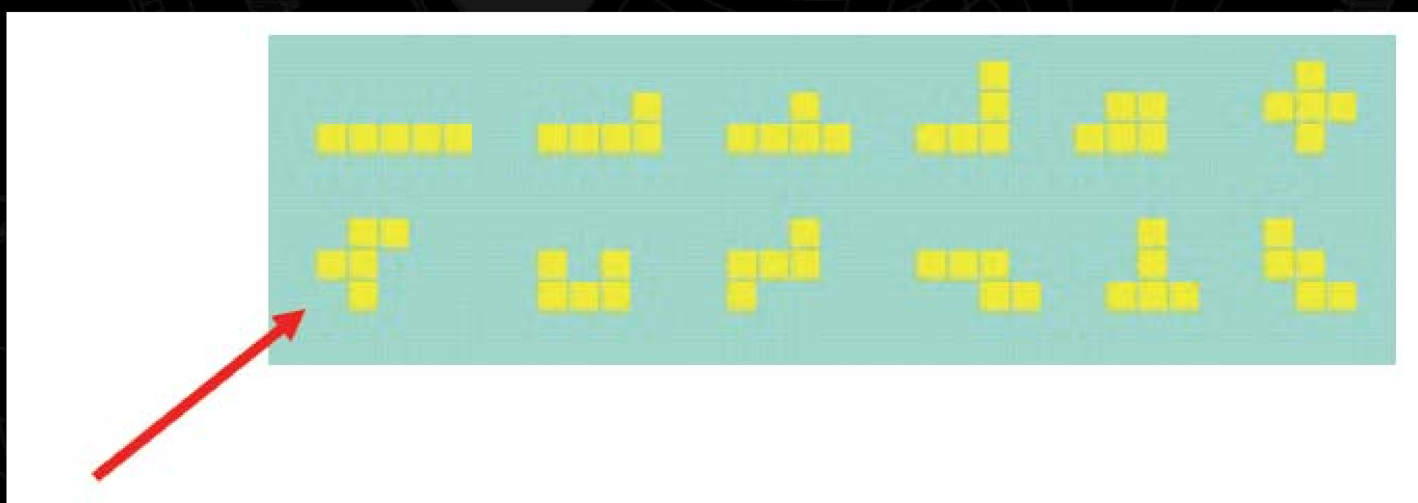


http://www.cogs.susx.ac.uk/users/andywu/gallery/ddlab_gallery.html

The R-Pentomino

Exhaustive study of all connected 5 cell initial states

All relatively uninteresting (quickly die or become Still Life), except for the r-pentomino which runs 1103 steps (at which point the “population” stops growing), by when it has become 6 gliders, 8 blocks, 4 beehives, 4 blinkers, 1 ship, 1 boat, and 1 loaf ☺

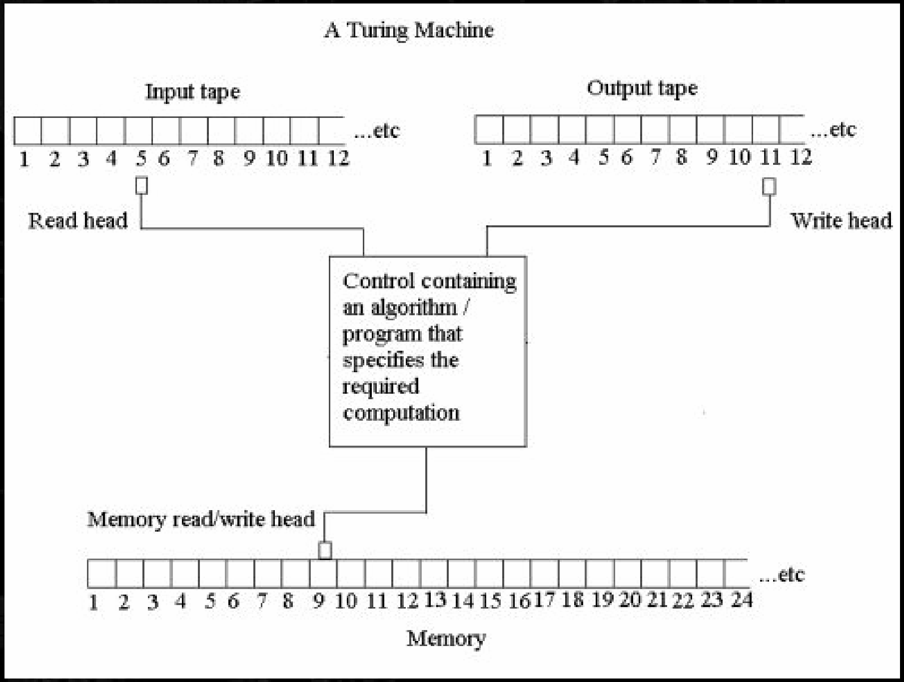


http://www.math.com/students/wonders/life/life.html

Game of Life: Universal Computation

U.C. means that there is the capability of computing anything that can be computed (e.g. computable numbers, transcendental numbers, computable functions).

Best known example is a UTM – an imaginary machine proposed in 1936 by A. M. Turing.



Example

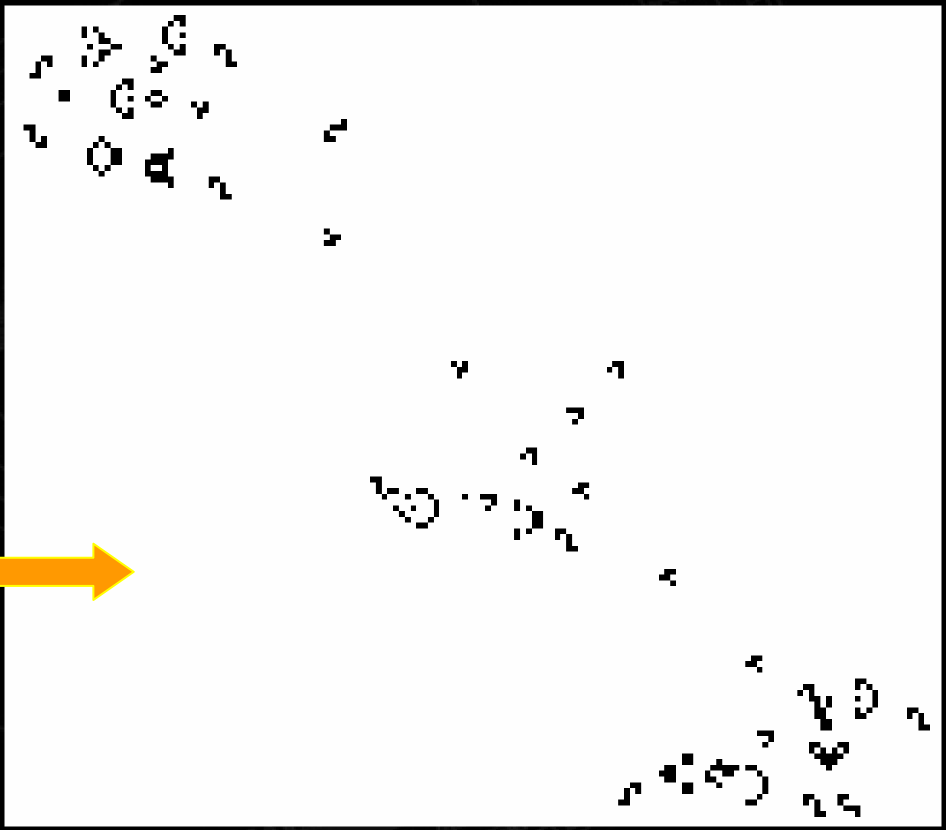
Input: Coins, bills, choice button

State: money entered so far

Output: Sprite, Coke, …

Logic Gates

- Glider gun to produce string of gliders
- In a stream, gliders as “1”s, gaps as “0”s
- Carefully arrange streams to intersect and annihilate, to produce “gates”
- NOT XOR gate example
- First step to building a computer in Life!



http://rendell-attic.org/gol/pictures/notxor.htm

Another 2-D CA: Langton's Ant

Langton's ant = 2-D CA with a very simple set of rules

Squares on a plane are colored variously either black or white. One square is arbitrarily identified as the "ant". The ant can travel in any of the four cardinal directions at each step it takes. The ant moves according to the rules below:

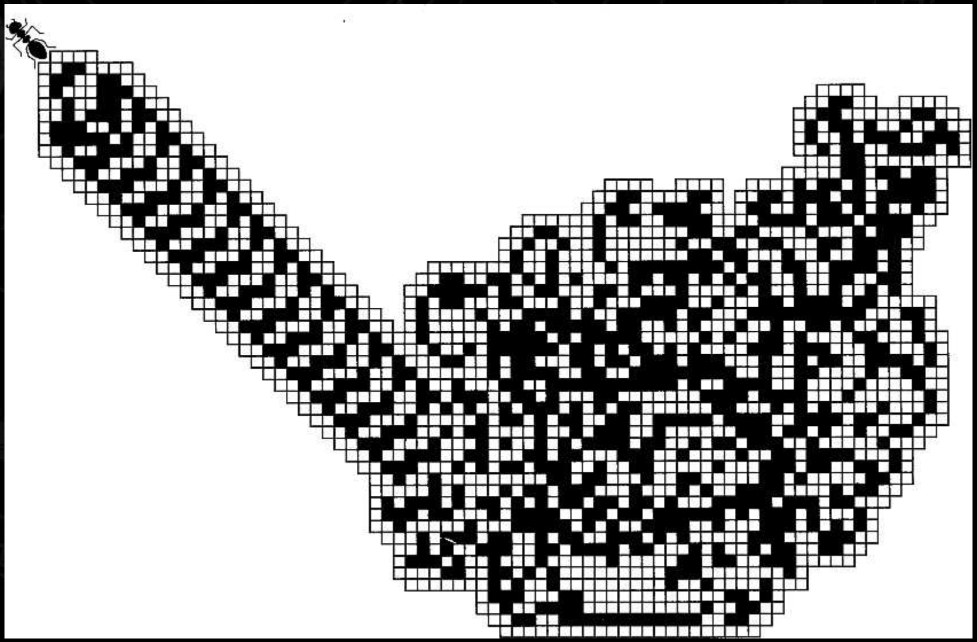
1) At a black square, turn 90° right, flip the color of the square, move forward one unit

2) At a white square, turn 90° left, flip the color of the square, move forward one unit

These simple rules lead to surprisingly complex behavior

Langton’s Ant (“Vant”)

The ant appears invariably to start building after <u>thousands of steps</u> (transient) a road of 104 steps that repeat indefinitely - regardless of the pattern you start off with. This suggests that the “highway” configuration is an attractor of Langton's ant.



NetLogo! + SW

Applications – Science, Technology, Art …

Cellular automata can be used to model complex systems using simple rules

Key features*

- divide problem space into cells
- each cell can be in one of several finite states
- cells are affected by neighbors according to rules
- all cells are affected simultaneously in a generation
- rules are reapplied over many generations

Forest Fire Model

Forest Fire Model

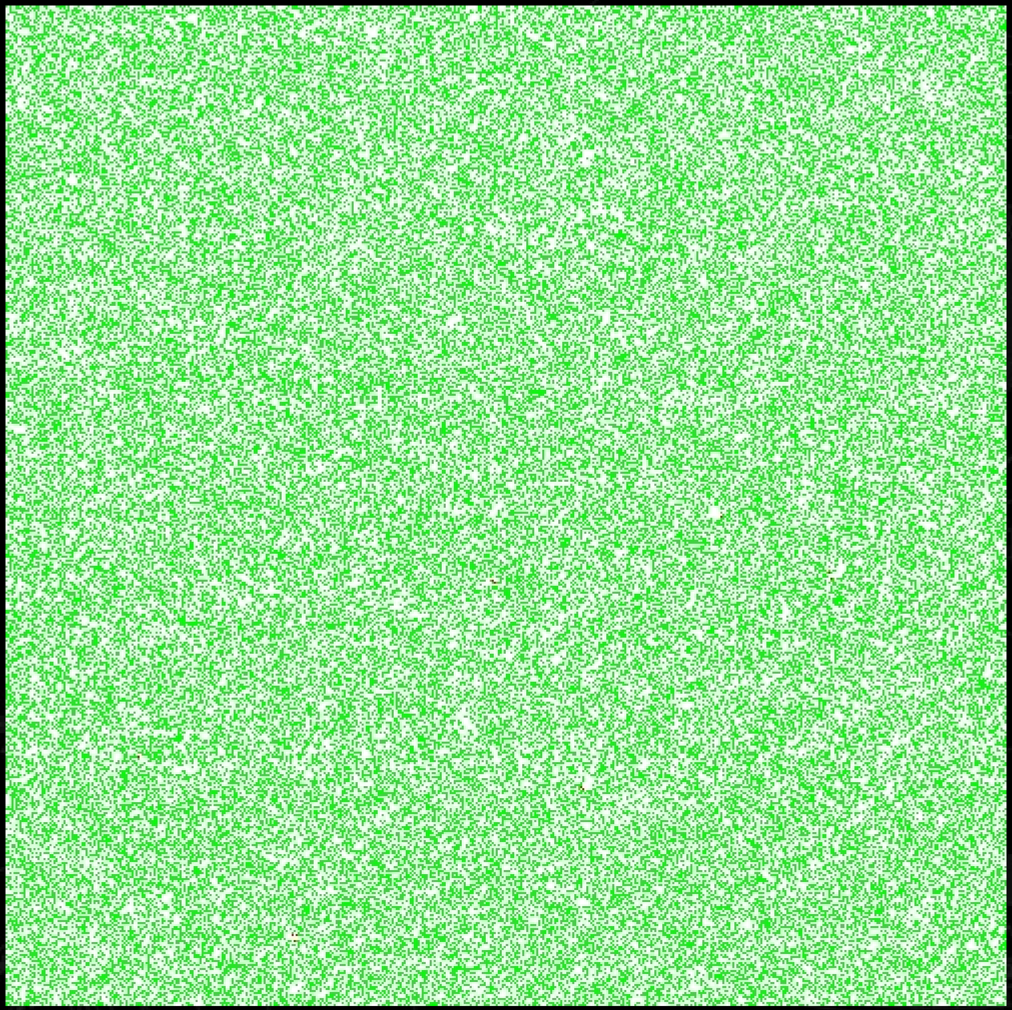
Forest Fire Model is a stochastic *3*-state cellular automaton defined on a *d*-dimensional lattice with *L* sites

Each site is occupied by a tree, a burning tree, or is empty

During each time step the system is updated according to the rules:

1) empty site (state 0) $\rightarrow$ tree (state 1): with the growth rate probability *p*
2) tree (state 1) $\rightarrow$ burning tree (state 2): with the lightning rate probability *f*, if no nearest neighbour is burning
3) tree (state 1) $\rightarrow$ burning tree (state 2): with the probability 1-*g*, if at least one nearest neighbour is burning, where *g* defines immunity.
4) burning tree (state 2) $\rightarrow$ empty site (state 0)

Simulation

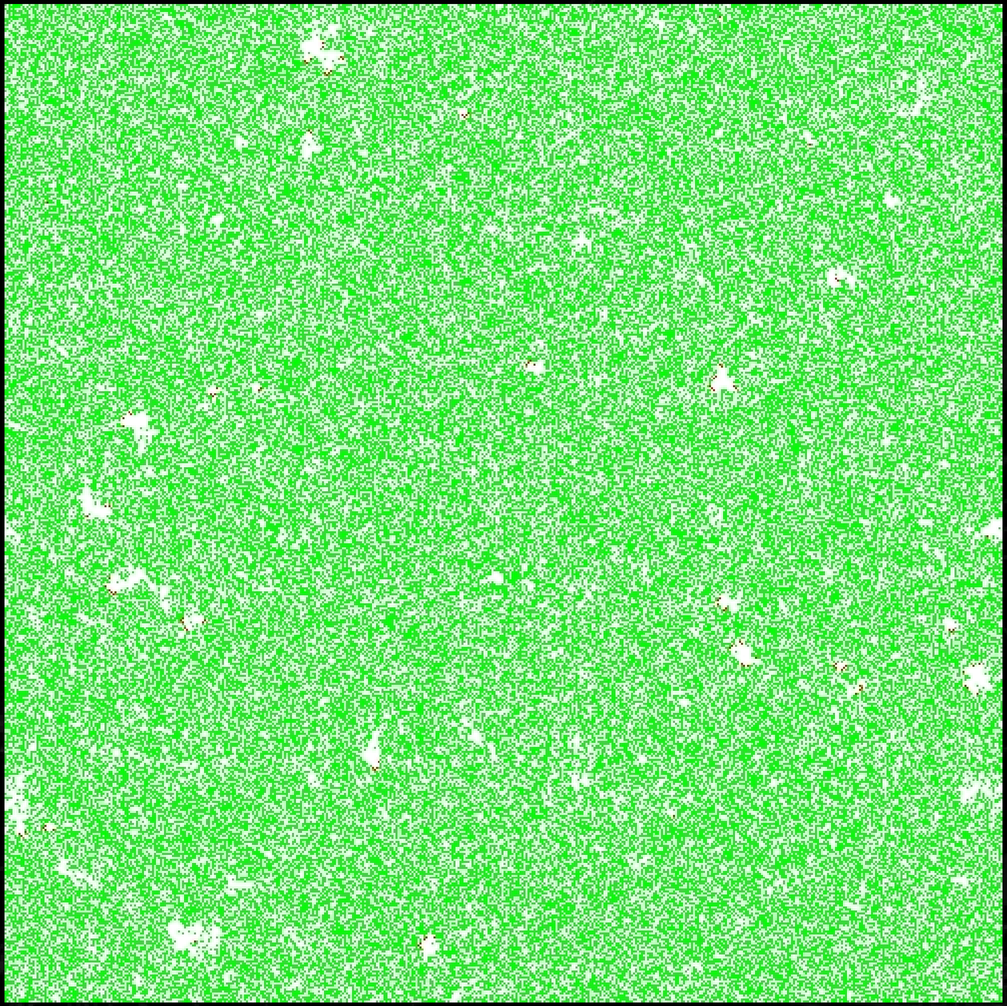


forest density = 45%

fire is not visible

average cluster is small in comparison to lattice size L

Simulation



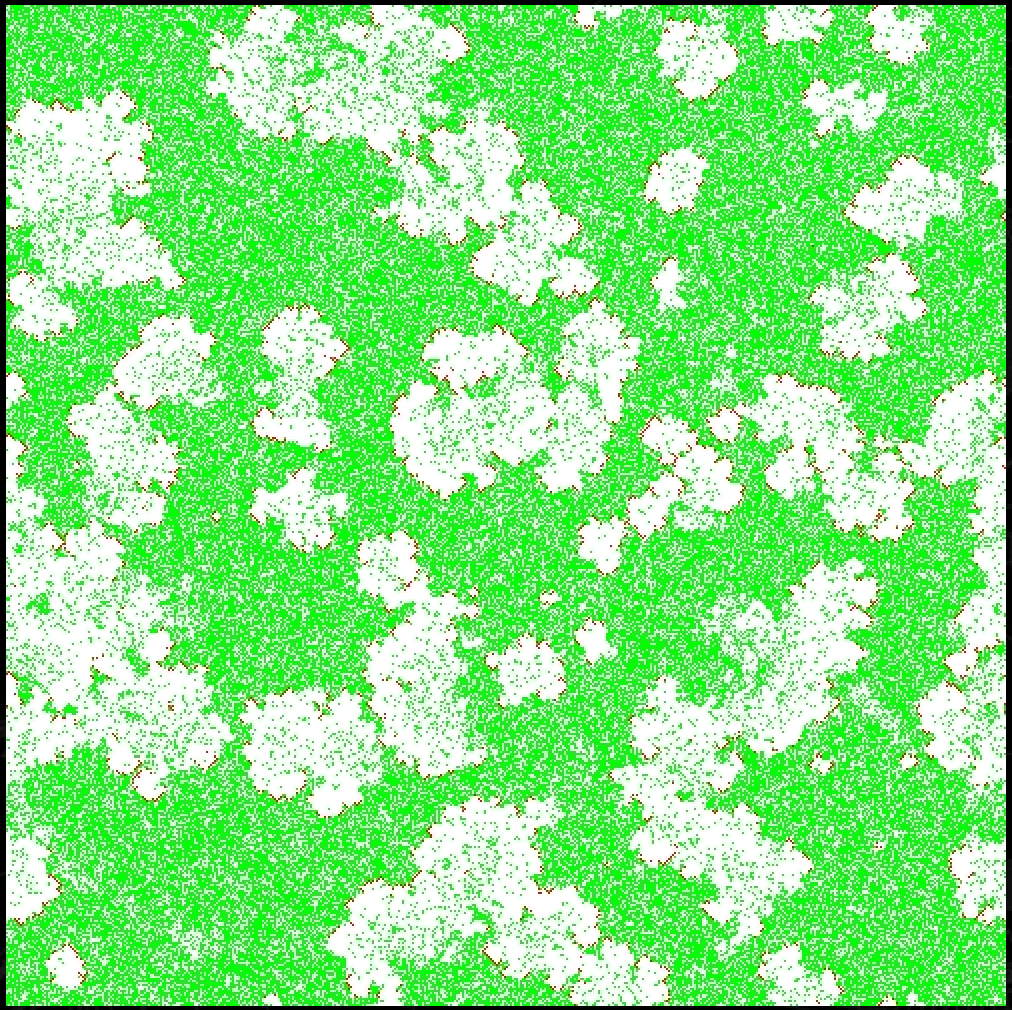
forest density 60%

first signs of fire

forest density reaches the critical value 59% - the percolation threshold for square lattice

the average cluster size goes to infinity for infinite lattice size

Simulation



fire spreads quickly burning down all connected tree clusters

a variety of global structures emerges

the whole process repeats and after some time forest reaches *the steady state* in which the mean number of growing trees equals the mean number of burning trees

Simulation

Time for NetLogo!

→ EarthScience → Fire

FFM and Relationship to Excitable Media

The FFM is closely related to excitable media, which comprise phenomena like spreading of diseases, oscillating chemical reactions, spiral galaxies, etc.

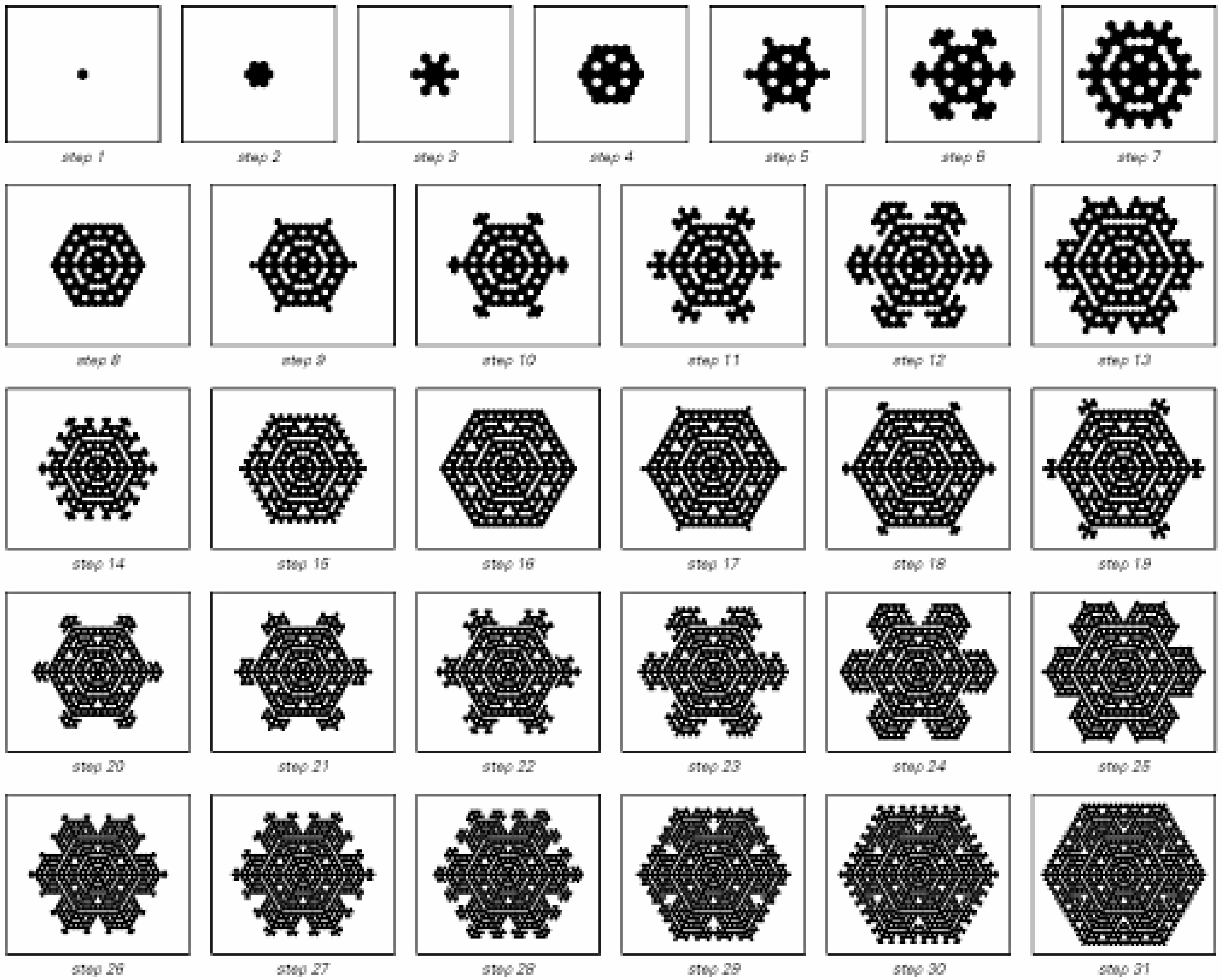
The system have 3 states:

1) quiescent (tree)
2) excited (burning tree)
3) refractory (empty side)

Excitation (fire front) spreads from one place to its neighbours if they are quiescent

After excitation, a refractory site needs some time to recover its quiescent state

Growth of Crystal – Snowflakes



The evolution of a cellular automaton in which each cell on a hexagonal grid becomes black whenever exactly one of its neighbors was black on the step before. This rule captures the basic growth inhibition effect that occurs in snowflakes. The resulting patterns obtained at different steps look remarkably similar to many real snowflakes.

(Wolfram, 2002)

Autocatalytic Chemical Sets

- Collection of chemicals that is self-catalyzing and therefore capable of highly nonlinear dynamics
- Of relevance for the origin of life
- Given chemical A, B and reagent C, suppose that A + B → C (slow process); catalyst D is a chemical that increase reaction rate by orders of magnitude: A + B + D → C (fast process); things get interesting when C catalyzes yet another reaction (say A + D → F)
- Molecules are located near each other → local interactions → can be modeled with a CA!

Parallel Particle Systems

Takai et al. (1995) “A cellular automaton model of particle motions and its applications” *The Visual Computer*, 11(5)

Spherical CA

J.J. Ventrella (2006)

http://www.ventrella.com/EarthDay/EarthDayAnimation.html

CA and Music

http://www-128.ibm.com/developerworks/java/library/j-camusic/

http://tones.wolfram.com/