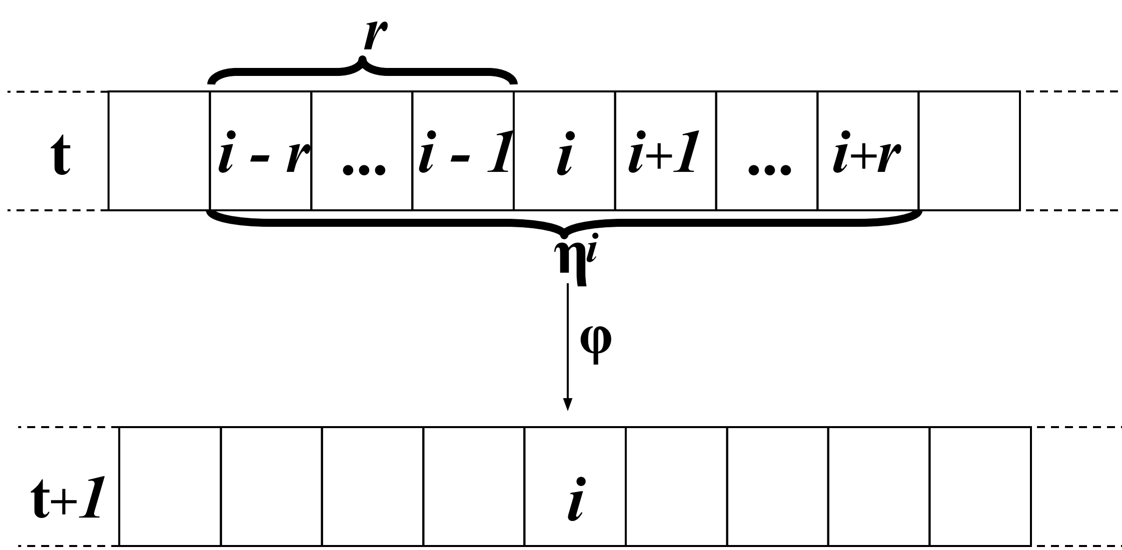
One-Dimensional CA Illustration



Example One-Dimensional CA
Rule 110

- The number of states, *k*=2.
- The alphabet $\Sigma = \{0,1\}$ $\quad |\Sigma| = k$
- The neighborhood η
- The output bit $s_{t+1}^{i} = \phi(\eta_{t}^{i})$
- For rule table ϕ

- The rule table ϕ for a radius r = 1, k=2 (binary CA):
  k^{2r+1} neighborhood configurations in

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

- In Wolfram notation this is rule 110 (base 10) because the output states are: 01101110 (base 2) = 110 base 10
- Rule 110 supports universal computation.

Wolfram Rule 110

0	0	0	→	0
0	0	1	→	1
0	1	0	→	1
0	1	1	→	1
1	0	0	→	0
1	0	1	→	1
1	1	0	→	1
1	1	1	→	0

The number of rules (rows) in the rule table
is the number of neighborhood configurations

$$(|\eta| = k^{2r+1})$$

where k is the number of states (k = 2 if binary)

The number of rows in the rule table for the simplest CA (k=2, r=1) is

$$k^{2r+1} = 8$$

The number of possible rule tables (all possible combinations of green values) is ??

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

0 1 1 0 1 1 1 0 = 0+ 2 + 4 + 8 + 0 + 32 + 64 = Rule 110
Read the bit string BOTTOM to TOP (most significant bit at left)

Wolfram Rule 110

0	0	0	→	0
0	0	1	→	1
0	1	0	→	1
0	1	1	→	1
1	0	0	→	0
1	0	1	→	1
1	1	0	→	1
1	1	1	→	0

The number of rules (rows) in the rule table
is the number of neighborhood configurations

$$(|\eta| = k^{2r+1})$$

where k is the number of states (k = 2 if binary)

The number of rows in the rule table for the simplest CA (k=2, r=1) is

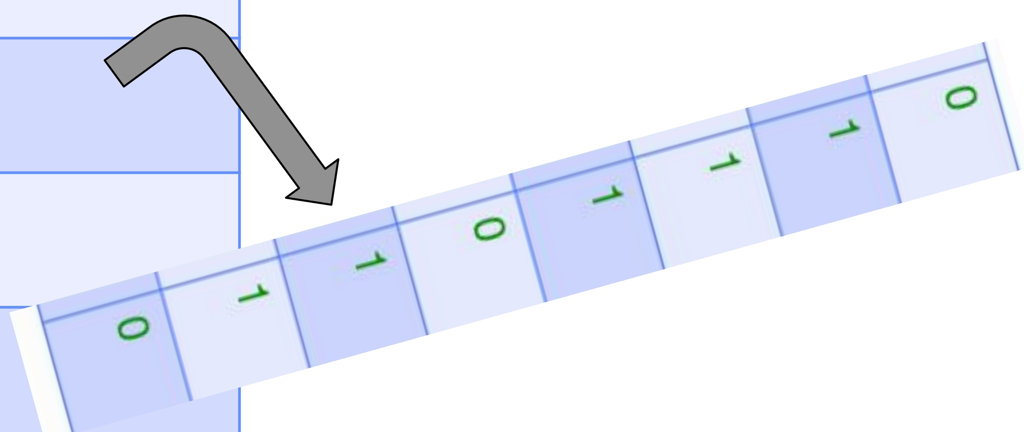
$$k^{2r+1} = 8$$

The number of possible rule tables (all possible combinations of green values) is 2^8 = **256**

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

0 1 1 0 1 1 1 0 = 0+ 2 + 4 + 8 + 0 + 32 + 64 = Rule 110
Read the bit string BOTTOM to TOP (most significant bit at left)

0	0	0	□	0
0	0	1	□	1
0	1	0	□	1
0	1	1	□	1
1	0	0	□	0
1	0	1	□	1
1	1	0	□	1
1	1	1	□	0



Rule 110

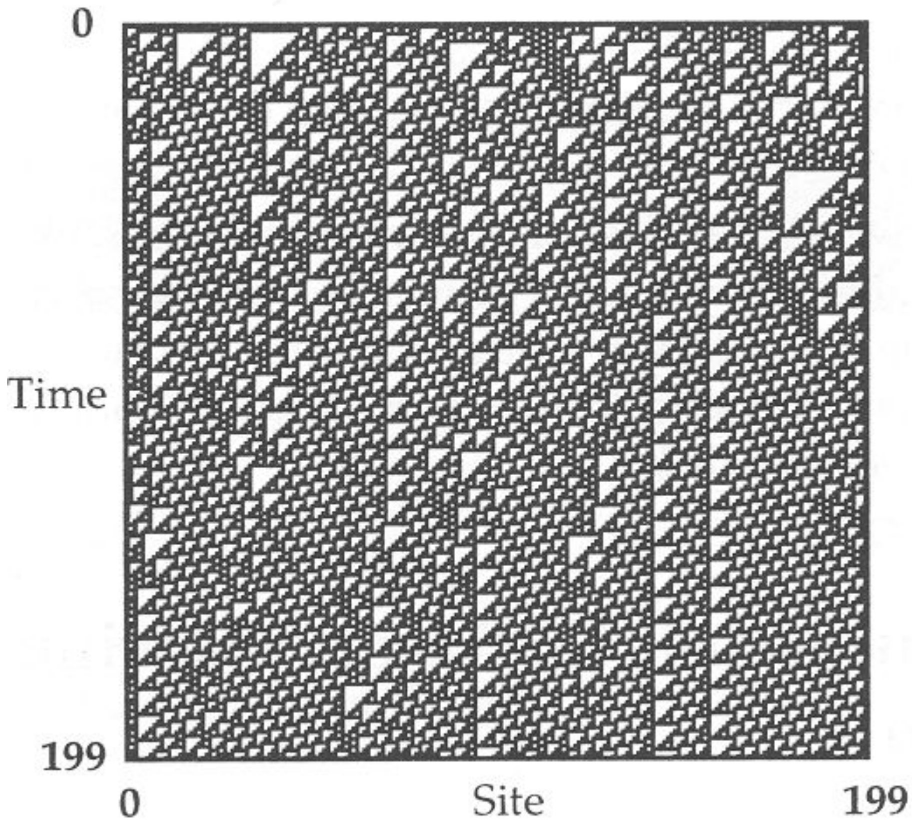
Read the bit string BOTTOM to TOP

111 is leftmost (most significant) bit

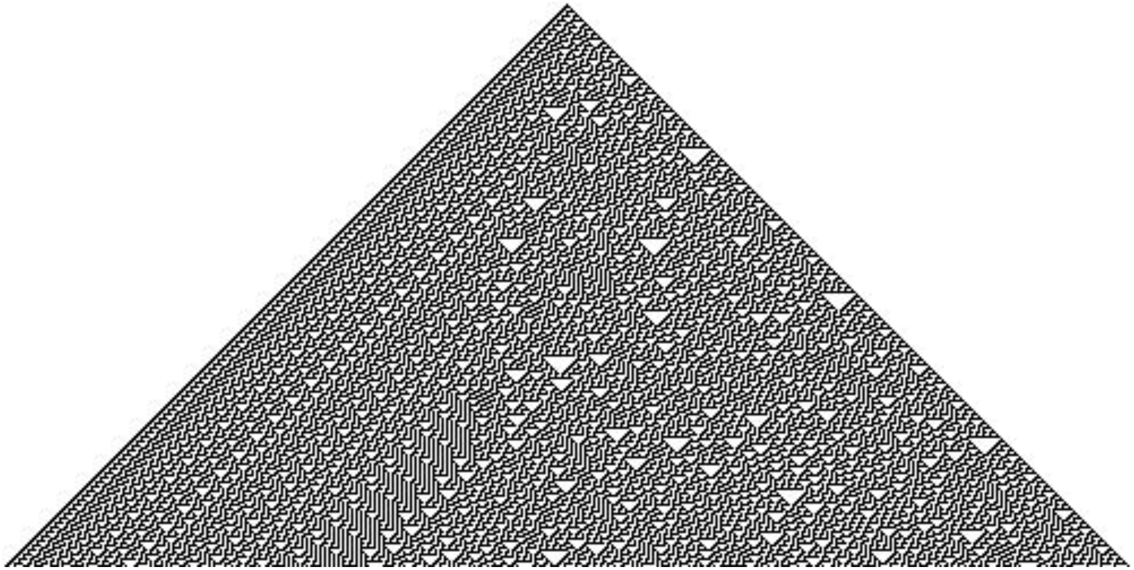
000 is the least significant bit

0 1 1 0 1 1 1 0 = 0+ 2 + 4 + 8 + 0 + 32 + 64 (adding numbers right to left - least significant bit first) = Rule 110

Rule 110 Space-Time Plot



Rule 30



current neighborhood	111	110	101	100	011	010	001	000
new state for center cell	0	0	0	1	1	1	1	0

Comments on Rule 30

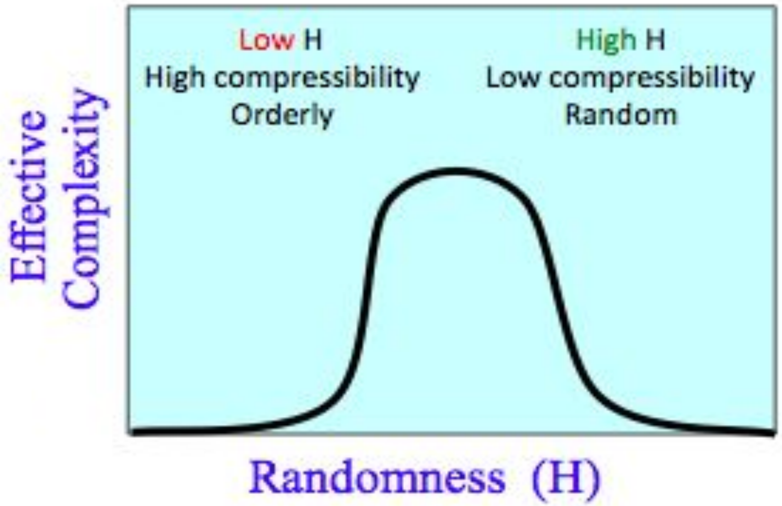
- Generates apparent randomness, despite being finite
- Wolfram uses the central column as a pseudo-random number generator in Mathematica
- Passes many tests for randomness, but many inputs produce regular patterns:
  - All zeroes
  - 00001000111000 repeated infinitely (try separating by 6 1s)

Wolfram's CA Classification

- Class I: Eventually every cell in the array settles into one state, never to change again. Analogous to
  - computer programs that halt after a few steps
  - dynamical systems that have fixed-point attractors
- Class II: Eventually the array settles into a periodic cycle of states
  - computer programs that execute infinite loops
  - dynamical systems that fall into limit cycles.
- Class III: The array forms "aperiodic" random-like patterns.
  - computer programs that are pseudo-random number generators (pass most tests for randomness, highly sensitive to initial condition).
  - Analogous to chaotic dynamical systems. Determinisitic but almost never repeat themselves, sensitive to initial conditions, embedded unstable limit cycles.

Wolfram's Classification cont.

- Class IV: The array forms *complex* patterns with localized structure that move through space and time:
  - Difficult to describe. Not regular, not periodic, not random.
  - Speculate that it is interesting computation.
- Hypothesis: The most interesting and complex behavior occurs in Class IV CA---``the edge of chaos''.
- Example: Rule 110



Wolfram Class I

Every cell in the array settles into one state

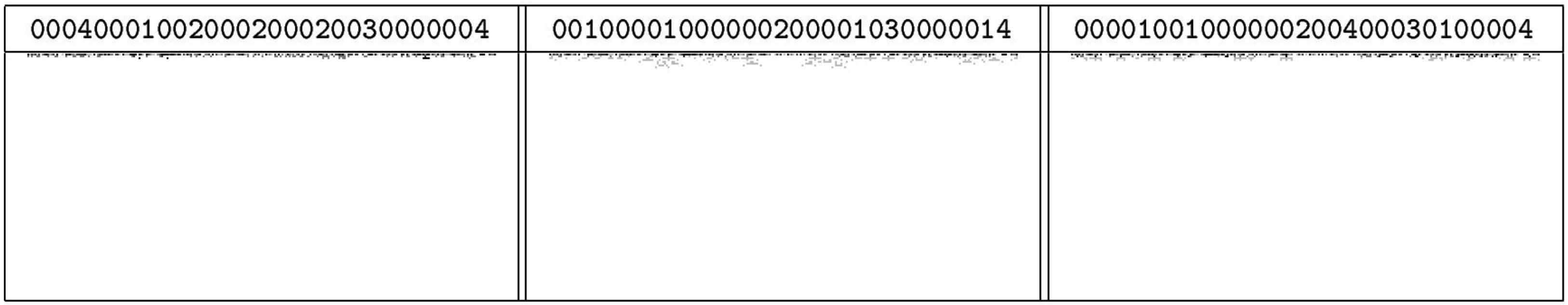
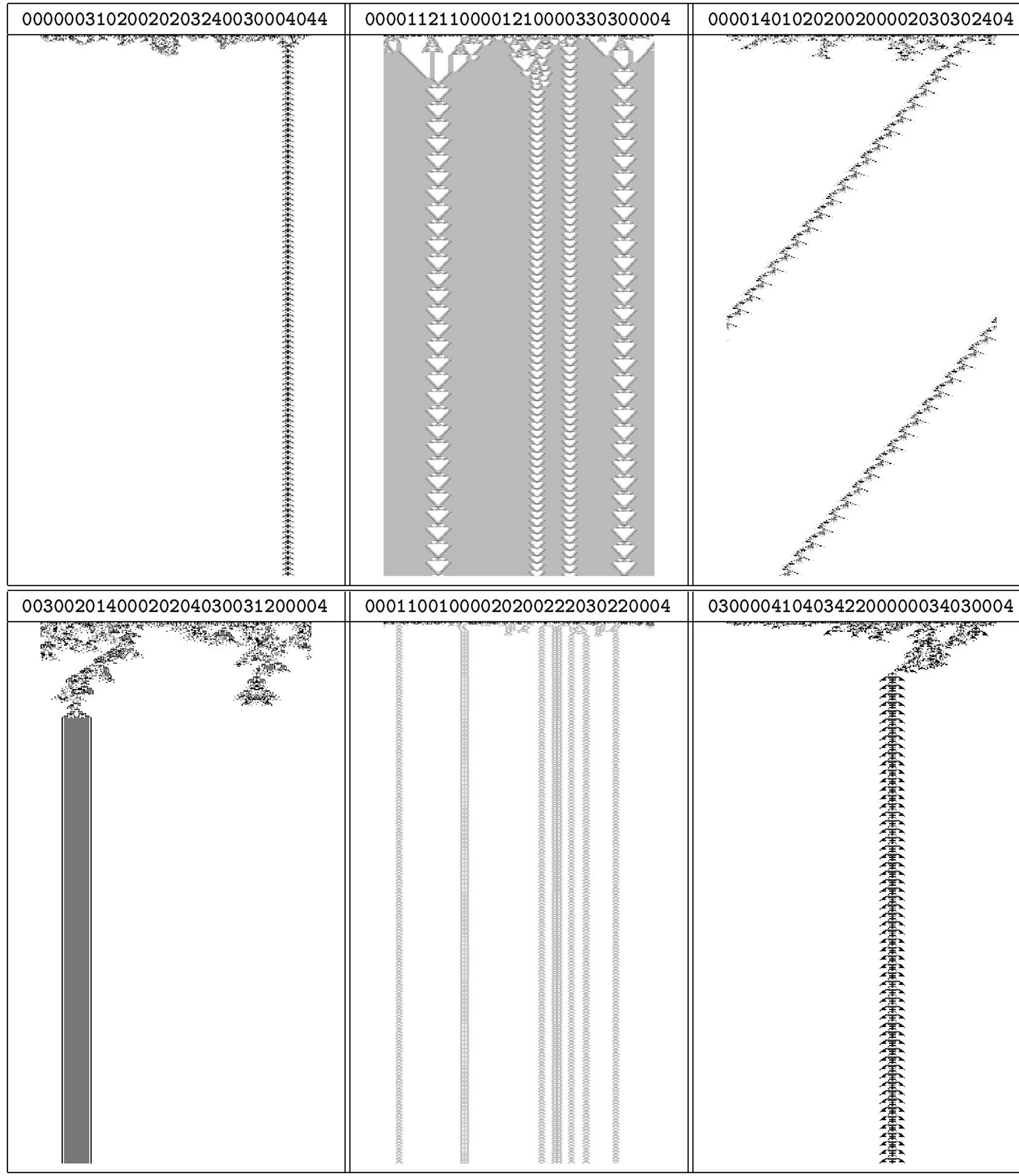


Figure 15.5 Examples of Wolfram's Class I

Wolfram Class II

Settles into a periodic cycle of states



Wolfram Class III

Forms “aperiodic” random-like patterns - analogous to chaos: deterministic but unpredictable without running the CA. Output is sensitive to initial input. Rule 30 is Class III.

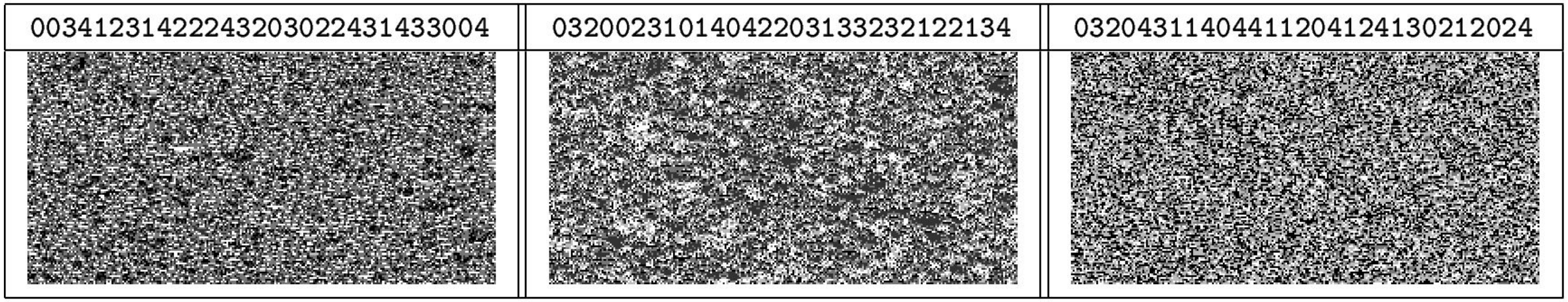


Figure 15.7 Examples of Wolfram’s Class III

Wolfram Class IV

complex patterns with localized structure that move through space and time. The edge of chaos.

Rule 110 is Class IV.

M

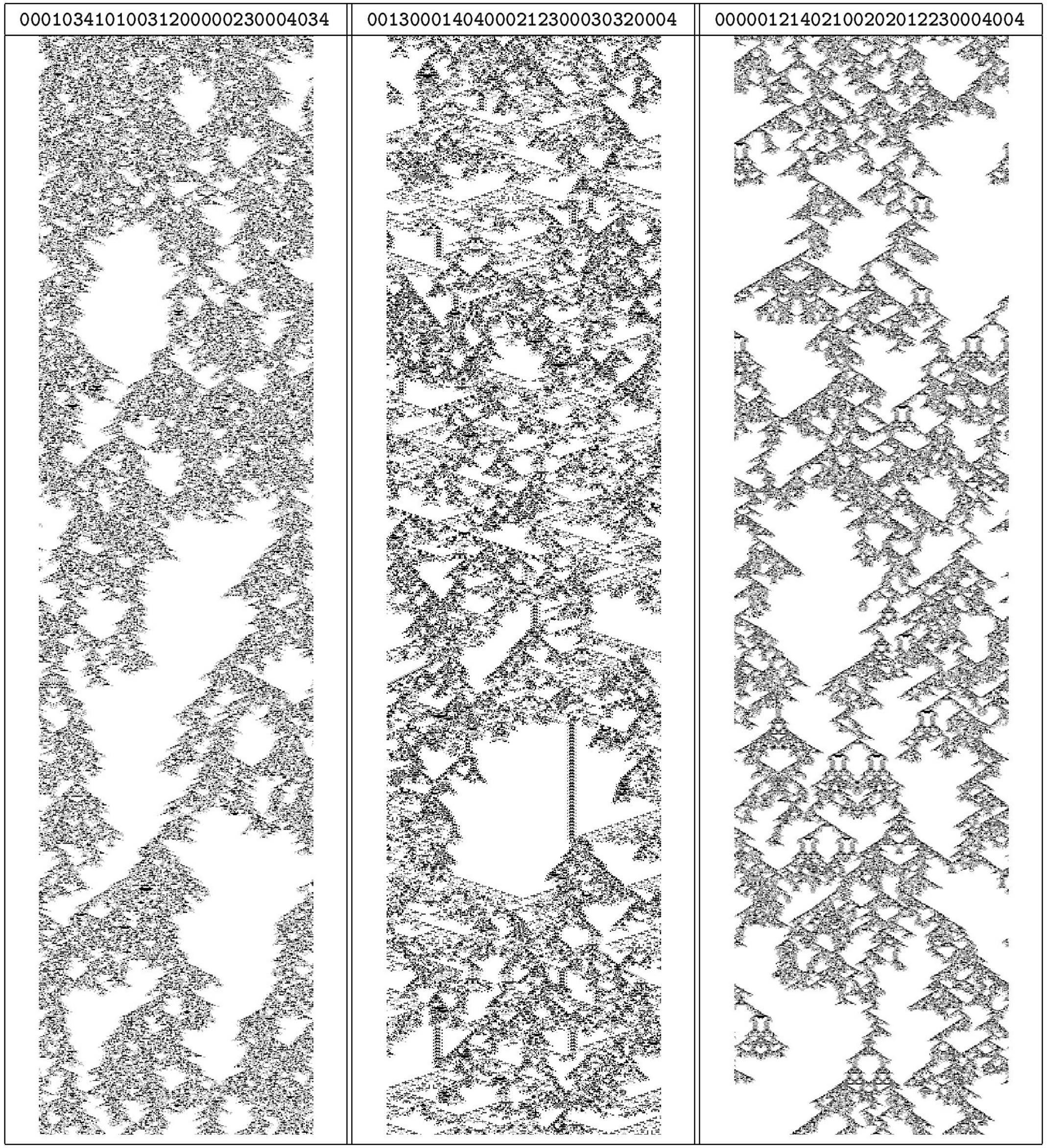


Figure 15.8 Examples of Wolfram's Class IV

Cellular Automata as models of complexity
Wolfram *Nature* 1984

http://twilightstarsong.blogspot.com/2009/03/shaping-future-stephen-wolfram-ray.html

- 4 classes
  - I: Fixed point
  - II: Limit cycle
  - III: Random or chaotic (unpredictable)
  - IV: Complex

- Wolfram's hypothesis:

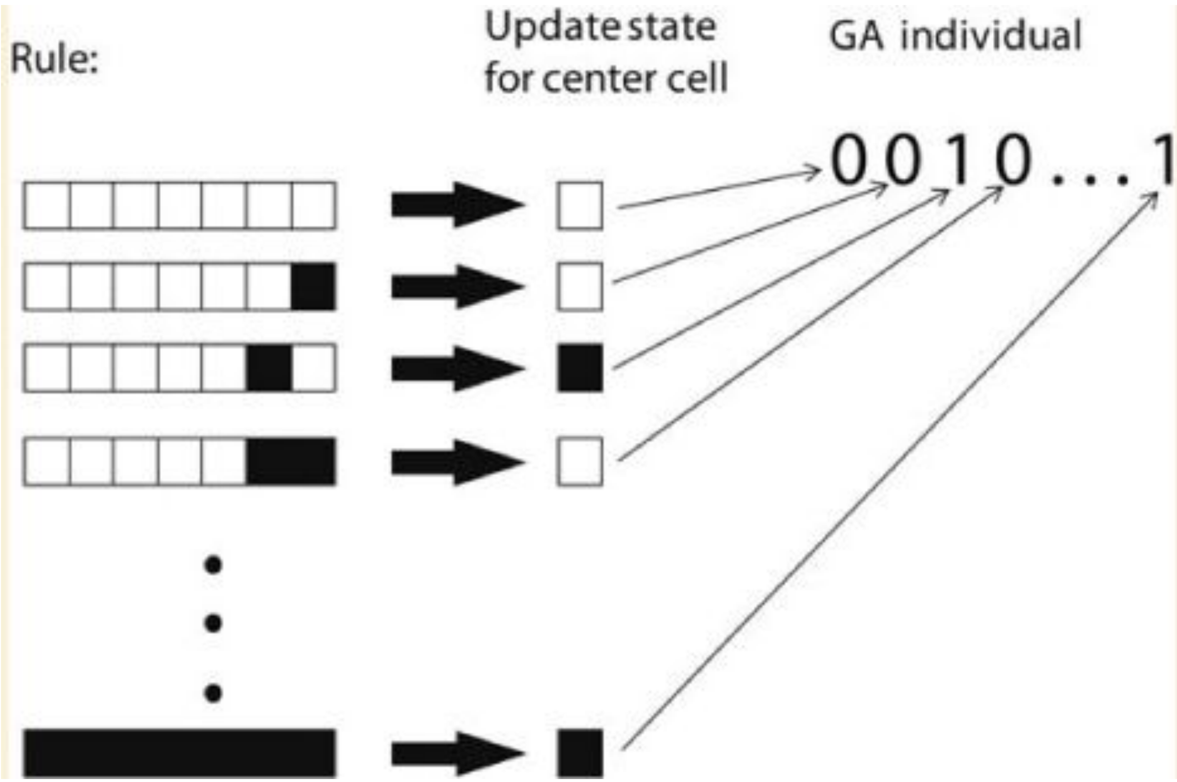
  Class IV supports universal computation
  - Long-term behavior undecidable and intractable
  - Simulation is necessary

In the density classification CA, the GKL rule appears to be a Class IV CA

Mitchell Chapter 11

- The majority classification task: the CA must compute whether its initial configuration is majority 1s or 0s
- 1 D CA with toroidal boundaries
- Must converge to all 1’s if initial configuration is majority 1’s
- Must converge to all 0’s if initial configuration is majority 0s
- Mitchell uses a GA to evolve rule sets to solve this problem

Mitchell Chapter 11

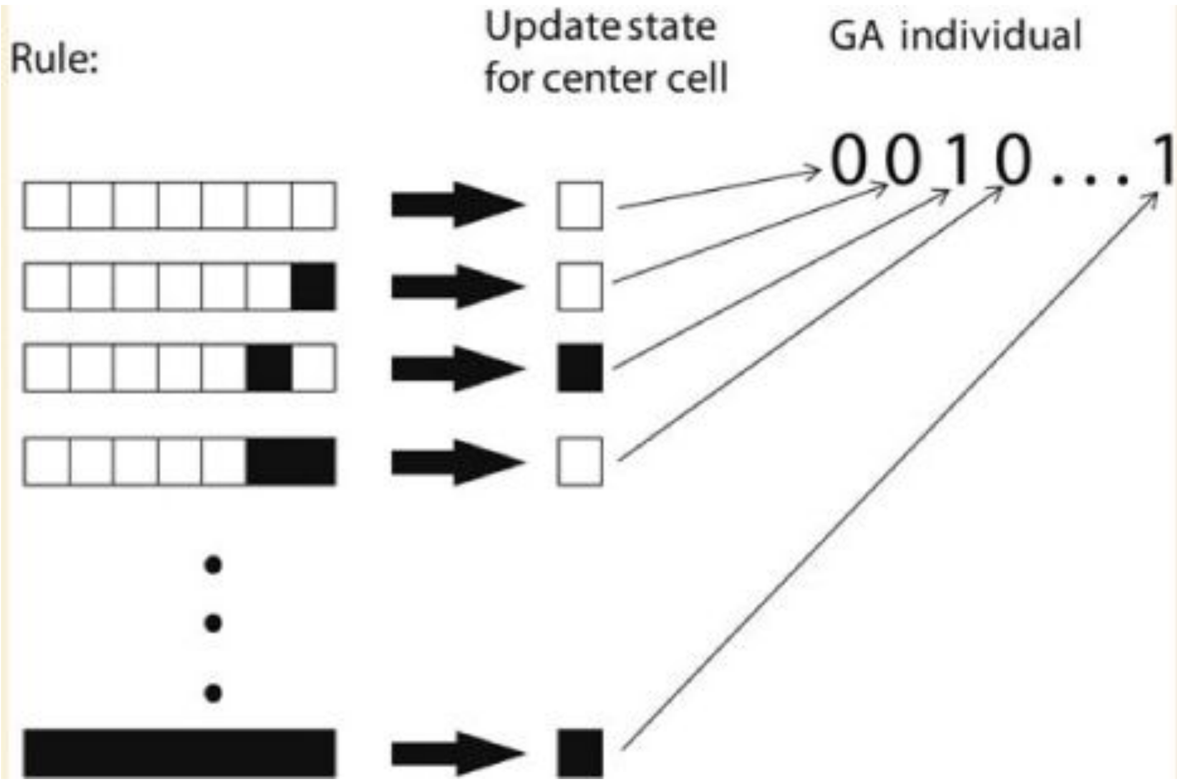


k = 2 (0 and 1)
r = 3, so neighborhood = 7

How many rows in each rule table?

How many possible rule tables is the GA searching through to find a rule table that accomplishes the density classification task?

Mitchell Chapter 11

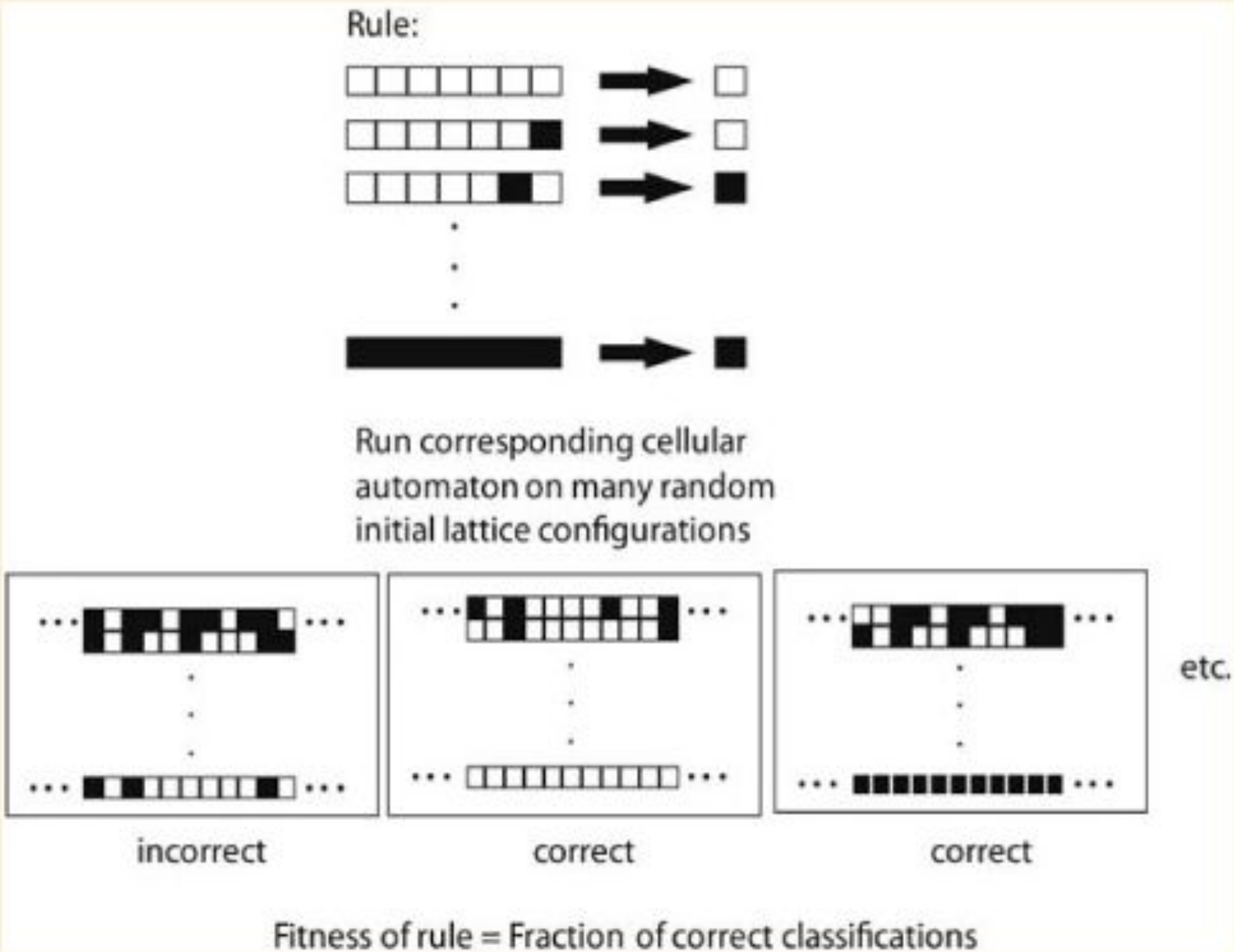


k = 2 (0 and 1)
r = 3, so neighborhood = 7

How many rows in each rule table?
$2^7 = 128$

How many possible rule tables is the GA searching through to find a rule table that accomplishes the density classification task?
2^{128}

$2^{10} = 1024$ (10^3)
2^{30} ~ 1 billion (10^9)
2^{60} ~ 1 billion billion (10^{18})
$2^{128} \sim 3\text{x}10^{38}$


Rule:
Run corresponding cellular automaton on many random initial lattice configurations
incorrect
correct
correct
etc.
Fitness of rule = Fraction of correct classifications

Exercise: Attempt to determine Global majority from local majority

Stand in a circle.
If your birthday is odd, raise 1 hand high
If it's even, keep hand down.

If your neighborhood (radius 1) is mostly odd, you become odd in the next time step.

If If your neighborhood is mostly even, you become odd in the next step.

Majority Rules Rule

What Wolfram Rule # is the majority vote rule for a radius 1 CA?

0	0	0	□	
0	0	1	□	
0	1	0	□	
0	1	1	□	
1	0	0	□	
1	0	1	□	
1	1	0	□	
1	1	1	□	

Read the bit string BOTTOM to TOP

111 is leftmost (most significant) bit

000 is the least significant bit

What Wolfram Rule # is the majority vote rule for a radius 1 CA?

0	0	0	□	0
0	0	1	□	0
0	1	0	□	0
0	1	1	□	1
1	0	0	□	0
1	0	1	□	1
1	1	0	□	1
1	1	1	□	1

Read the bit string BOTTOM to TOP

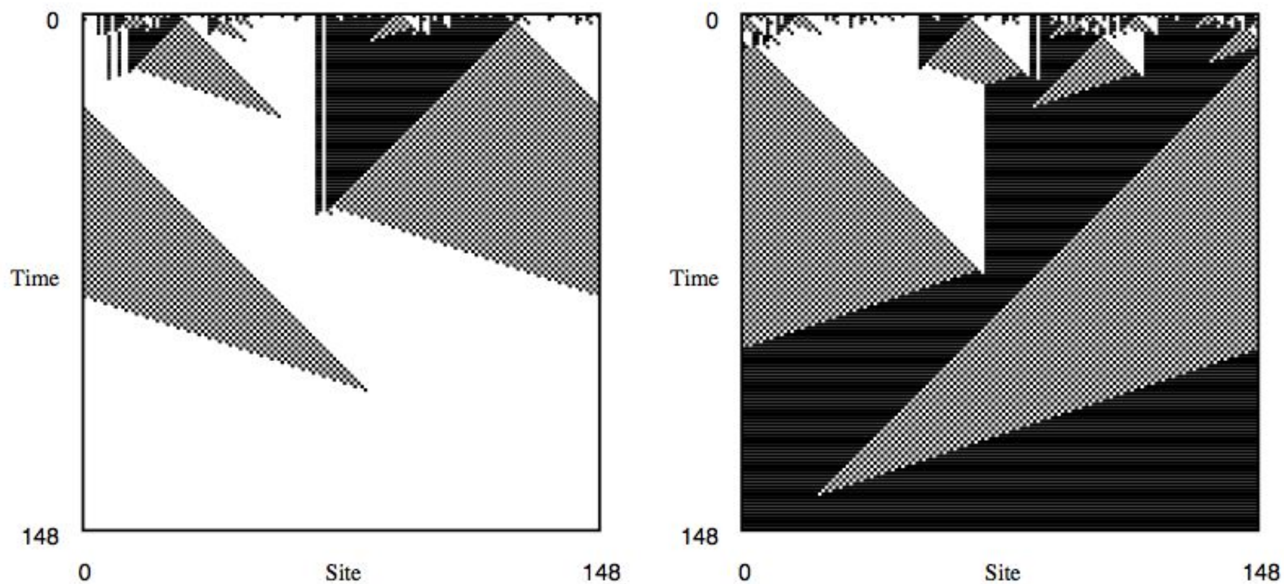
111 is leftmost (most significant) bit

000 is the least significant bit

11101000 = 232

The GKL Rule

- λ is the fraction of 1's in the rule table
- ρ_0 is the fraction of 1's in the initial condition
- correct classification is going to all white if $\rho_0 < ½$ black if $\rho_0 > ½$



(a) (b)

Figure 1: Two space-time diagrams for the binary-state Gacs-Kurdyumov-Levin CA. $N = 149$ sites are shown evolving, with time increasing down the page, from two different ICs over 149 time steps. Here cells with state 0 are white, and cells with state 1 are black. In (a), $\rho_0 \approx 0.48$, and in (b), $\rho_0 \approx 0.52$. Notice that by the last time step the CA has converged to a fixed pattern of (a) all 0s and (b) all 1s. In this way the CA has classified the ICs according to whether $\rho_0 > 1/2$ or $\rho_0 < 1/2$.

The GKL rule table

$$s_{t+1}^i = \phi(\eta_t^i) = \begin{cases} \text{majority}[s_t^i, s_t^{i-1}, s_t^{i-3}] & \text{if } s_t^i = 0 \\ \text{majority}[s_t^i, s_t^{i+1}, s_t^{i+3}] & \text{if } s_t^i = 1 \end{cases}$$

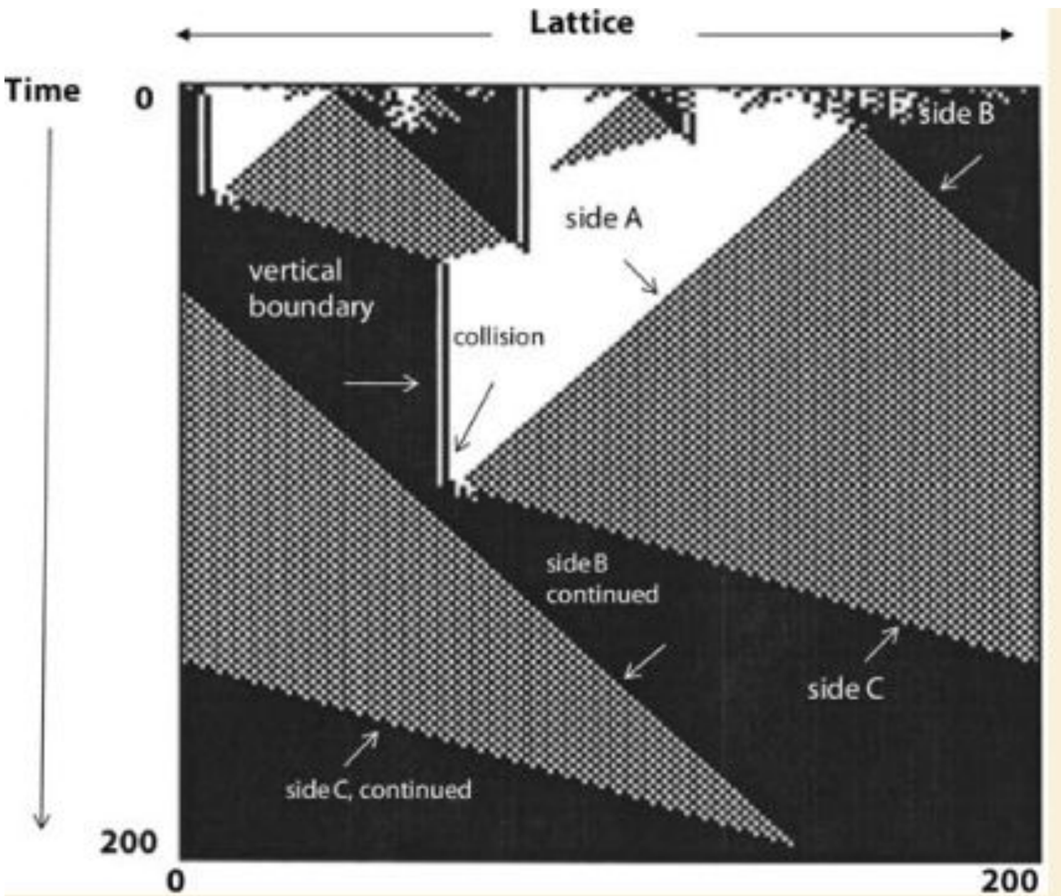
this rule says that for each neighborhood #i of seven adjacent cells, if the state of the central cell is 0, then its new state is decided by a majority vote among itself, its left neighbor, and the cell three sites to the left. λ = 1/2

The transition states of the rule table:

0000000

$$\mathbf{C} \;=\; \begin{array}{l} 00000000010111110000000001011111 \\ 00000000010111110000000001011111 \\ 00000000010111111111111101011111 \\ 00000000010111111111111101011111 \end{array}$$

1111111



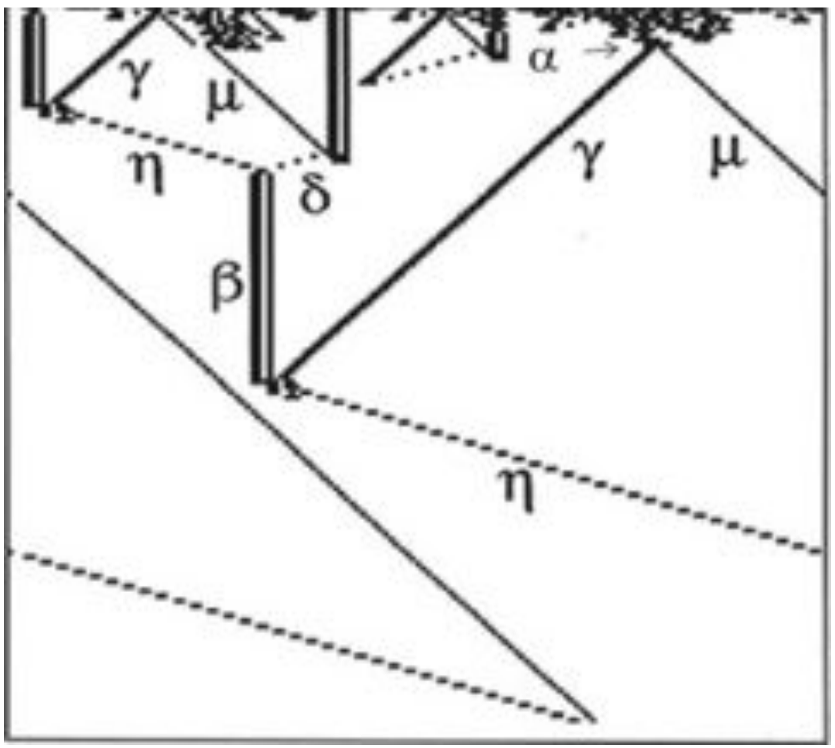
Whenever a black region on the left meets a white region on the right, there is always a vertical boundary.

Whenever a white region on the left meets a black region on the right, a checkerboard triangle forms.

Sides A and B of the growing checkerboard region travel at the same velocity

Side A travels southwest until it collides with the vertical boundary. Side B just misses colliding with the vertical boundary on the other side. This means that side A had a shorter distance to travel.

Mitchell, Melanie. Complexity: A Guided Tour



γ
μ
α →
η
δ
γ
μ
β
η

Information transfer in GKL

Typical space-time behaviors of the GKL rule for ICs with $\rho_0 < \rho_c$ and $\rho_0 > \rho_c$ were shown in Figure 1. It can be seen that, although the patterns eventually converge to fixed points, there is a transient phase during which a spatial and temporal transfer of information about local regions takes place. This local information interacts with other local information to produce the desired final state. Very crudely, the GKL rule successively classifies "local" densities with a locality range that increases with time. In regions where there is some ambiguity, a "signal" is propagated. This is seen either as a checkerboard pattern propagated in both spatial directions or as a vertical white-to-black boundary. These signals indicate that the classification is to be made at a larger scale. Note that regions centered about each signal locally have $\rho = \rho_c$. The consequence is that the signal patterns can propagate, since the density of patterns with $\rho = \rho_c$ is neither increased nor decreased under the rule.

In this way, local information processing at later times classifies larger patches of the IC. In a simple sense, this summarizes the rule's "strategy" for performing the computational

GKL errors are near $\rho_0 = 0.5$

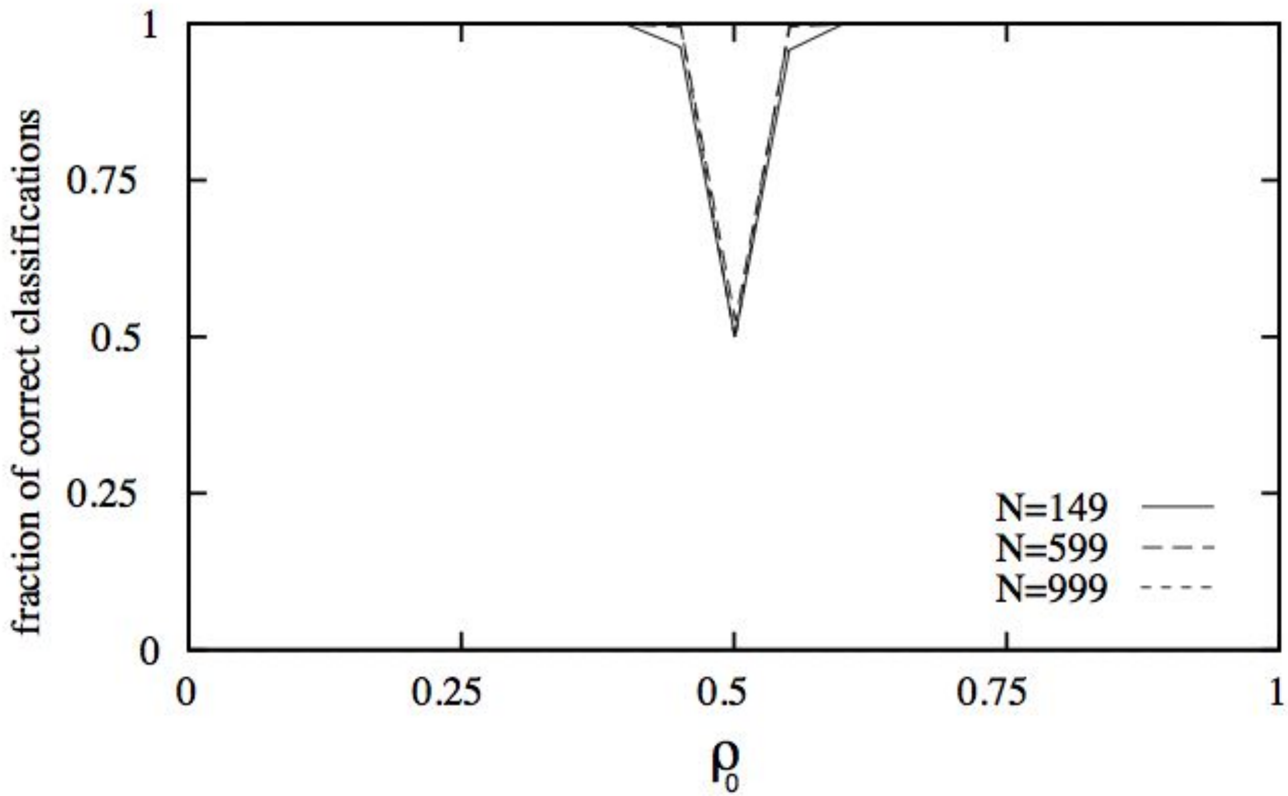


Figure 2: Experimental performance of the GKL rule as a function of ρ_0 for the $\rho_c = 1/2$ task. Performance plots are given for three lattice sizes: $N = 149$ (the size of the lattice used in the GA runs), 599, and 999. Note that the $N = 599$ and $N = 999$ curves are almost indistinguishable. (This figure differs slightly from Figure 4 in [61], since 21 density bins were used there.)

Checkerboard pattern transfers information about the density of black or white cells in local regions
Best evolved solutions are *almost* as good as the GKL rulset

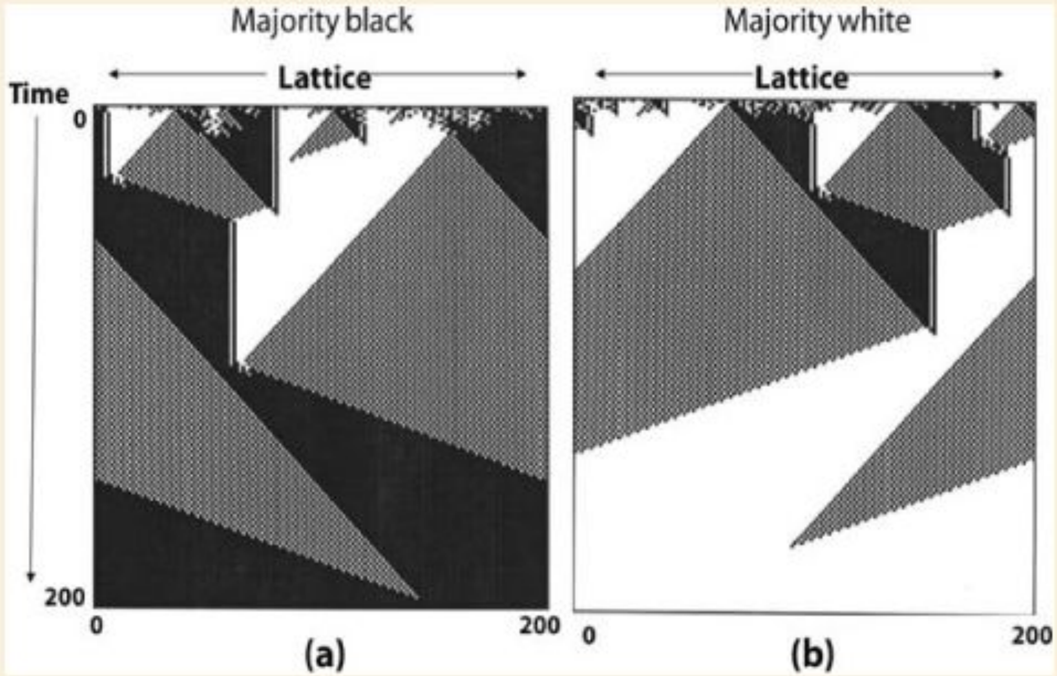


FIGURE 11.4. Space-time behavior of one of the best-performing evolved cellular automaton rules for the majority classification task. In (a), the initial configuration contains a majority of black cells and the cellular automaton iterates to a fixed configuration of all black. In (b), the initial configuration contains a majority of white cells and the cellular automaton iterates to a fixed configuration of all white. (Figure adapted from Mitchell, M., Crutchfield, J. P., and Das, R., Evolving cellular automata to perform computations: A review of recent work. In *Proceedings of the First International Conference on Evolutionary Computation and Its Applications (EvCA '96)*. Moscow, Russia: Russian Academy of Sciences, 1996.)

CA & GA specifications

(Useful Methods to list in a Table in order to replicate this study…)

- Elitism (top 20)
  - Other 80 rules are mutations (2%) & recombinations of elites
  - Elites are re-evaluated on new strings each generation
- Performance fitness: all 0s or 1s, **no partial credit**
- 1-point Crossover
- N = 149 (width of CA & length of bitstring for each IC)
- M = 320 (# of timesteps, height of CA, only 149 shown)
- I = 100 (# of IC), **uniform distribution over % of 0s** $\rho_0 \in [0.0, 1.0]$
- GA population size = 100
- GA generations = 100 xc

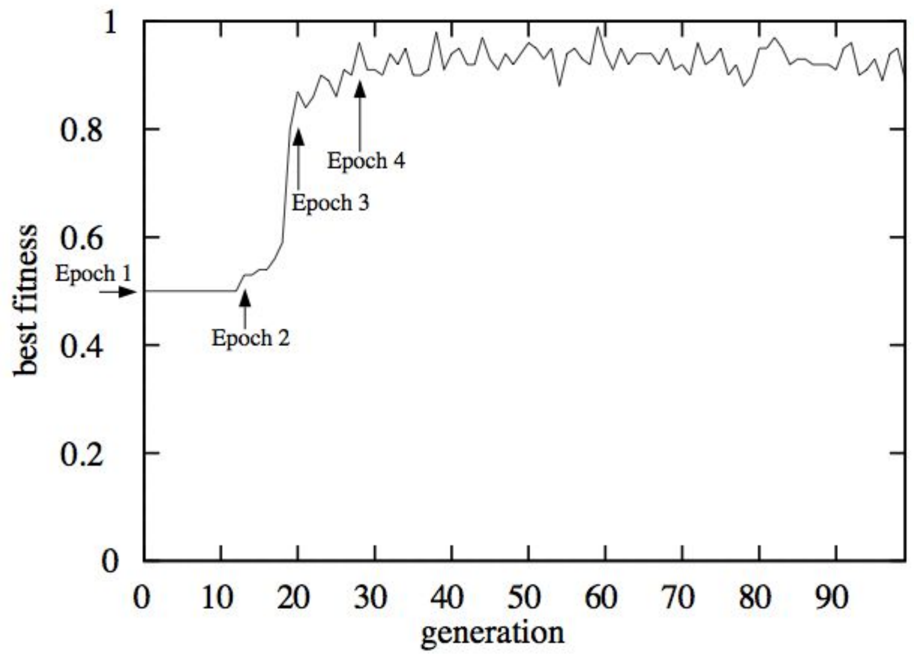
It should be pointed out as an aside that sampling ICs with uniform distribution over $\rho \in [0.0, 1.0]$ is highly biased with respect to an unbiased distribution of ICs, which is binomially distributed over $\rho \in [0.0, 1.0]$, and very strongly peaked at $\rho = 1/2$. However, preliminary experiments indicated a need for such a biased distribution in order for the GA to make progress in early generations. As we will discuss below, this biased distribution turns out to impede the GA in later generations because, as increasingly fitter rules are

What are “Epochs” in this study?

- Epochs are evolutionary transitions

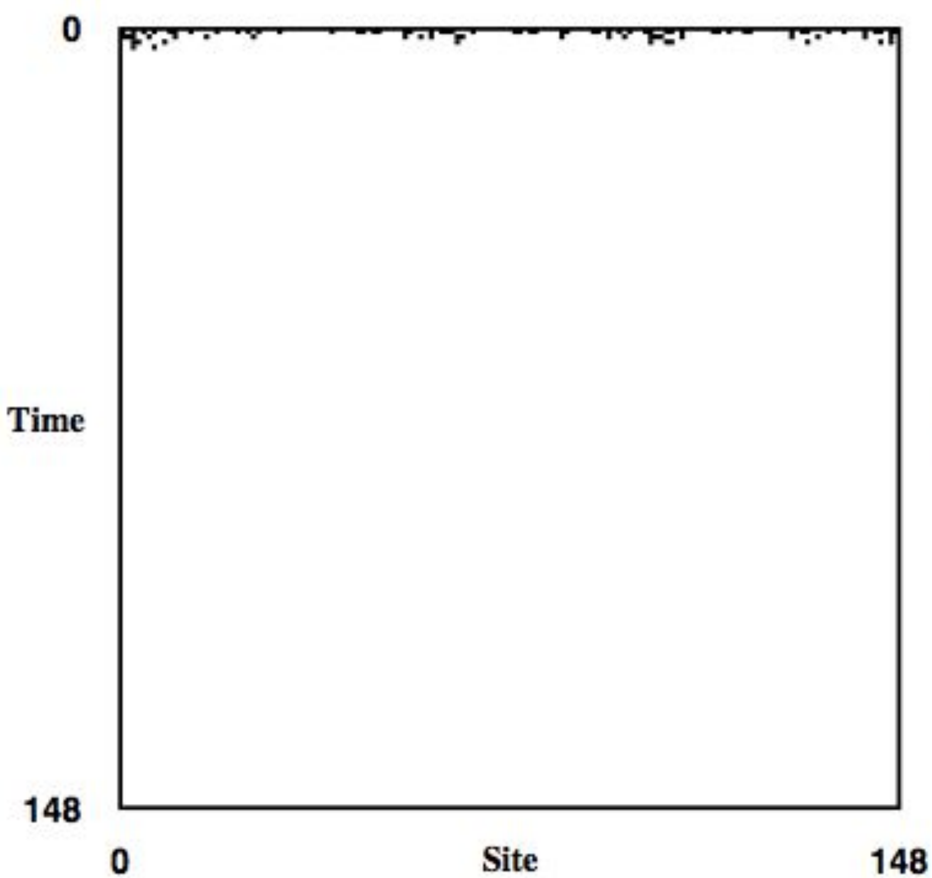
- Epoch 1: go to all 0s or all 1s regardless of input
- Epoch 2: keep choice in Epoch 1, but switch to all 1’s or 0’s in extreme cases
- Epoch 3: expand blocks
  – gives cover to allow drift for extreme cases
- Epoch 4: refine Epoch 3

(a)

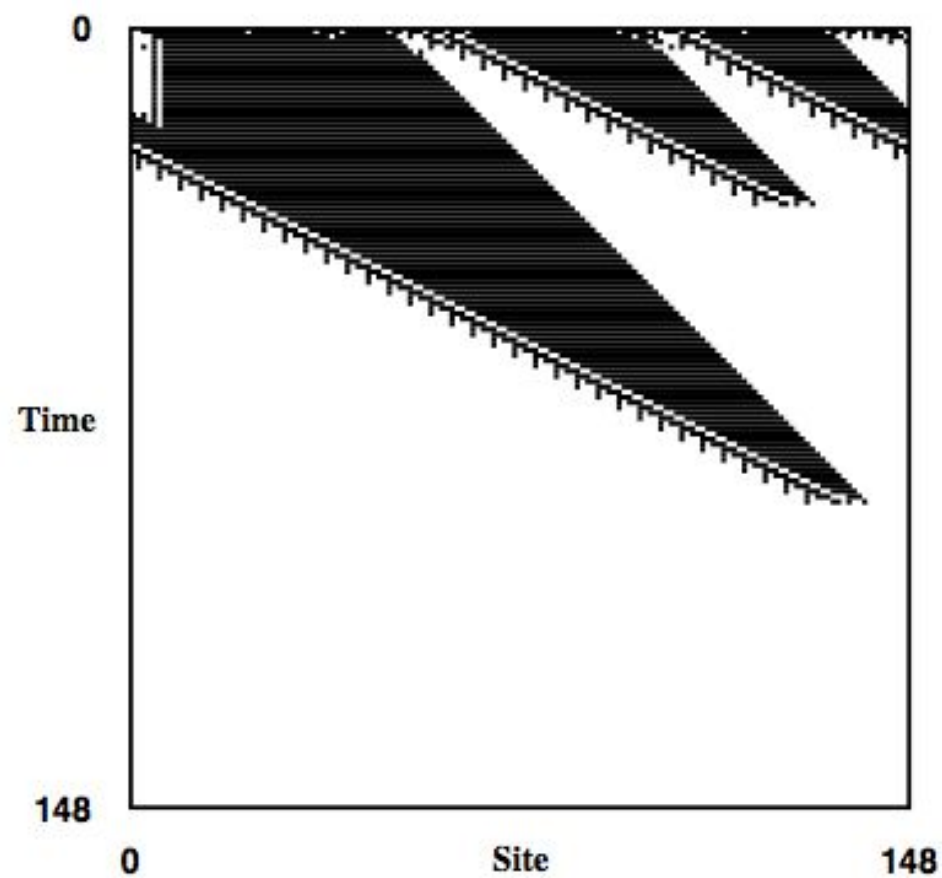


Epochs break symmetries increasing short term fitness but hindering discovery of highest fitness solutions

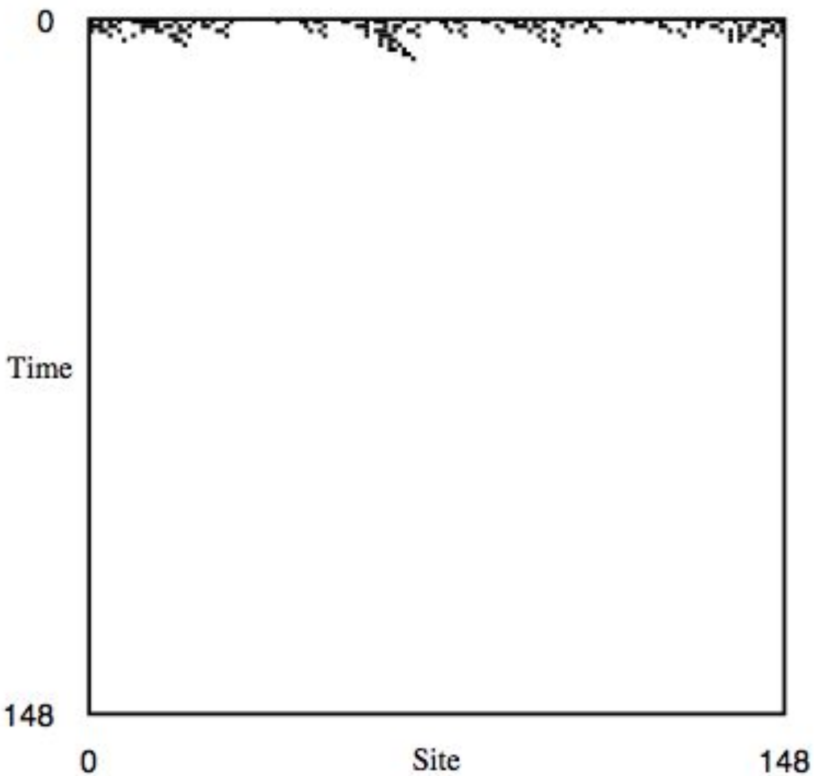
Epoch 1 & Epoch 2 for low p_0

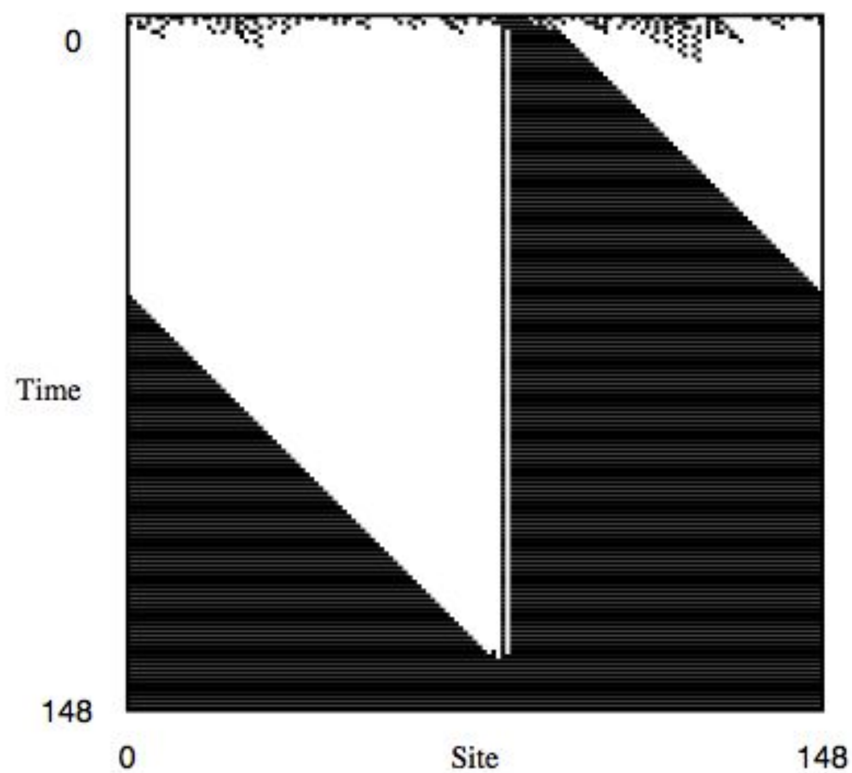


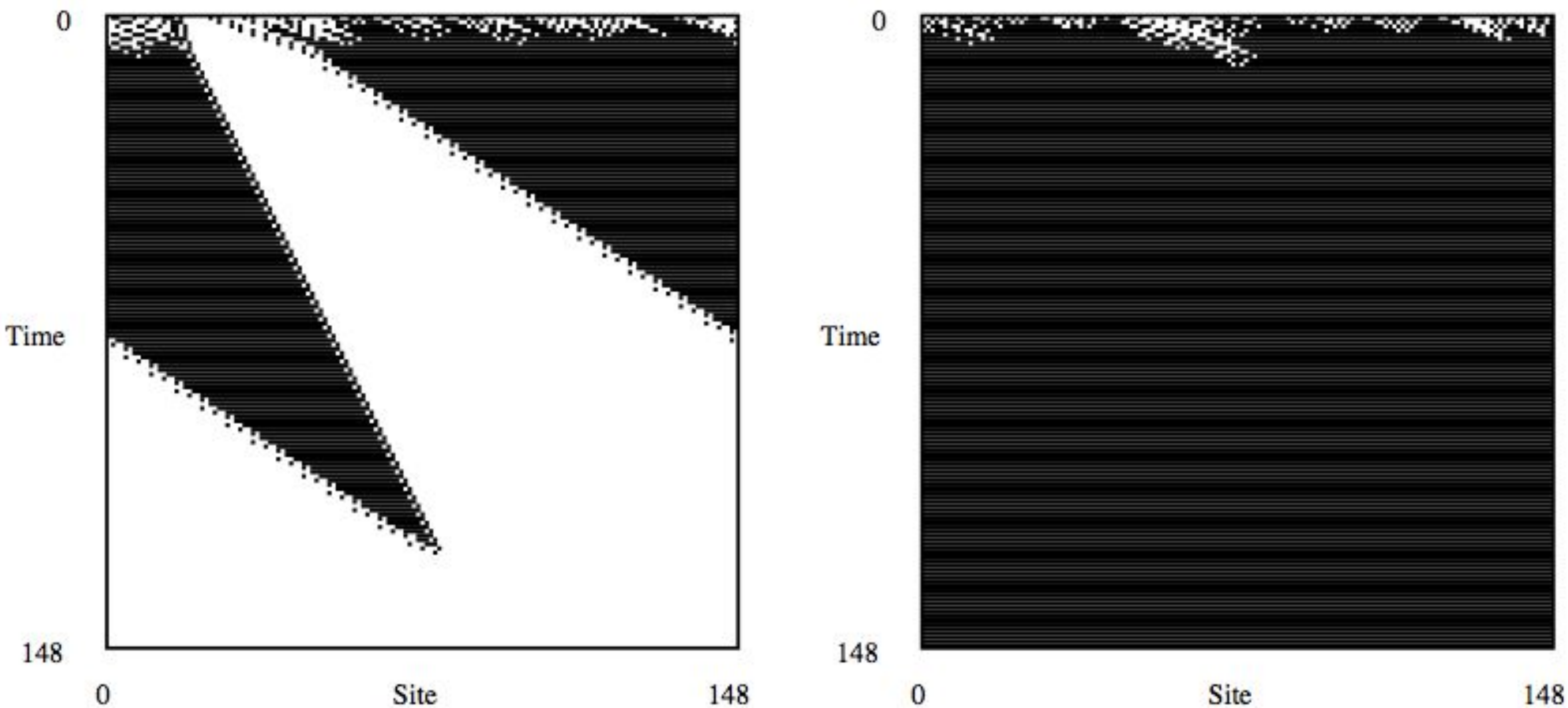
Epoch 2 for higher $p_{0,}$ still below ½



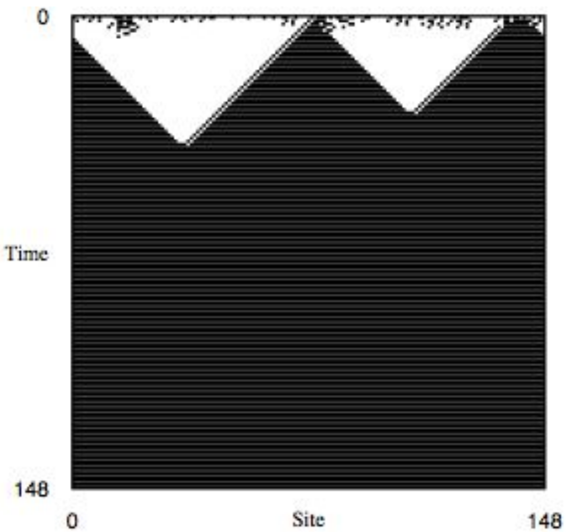
Epoch 3 & 4



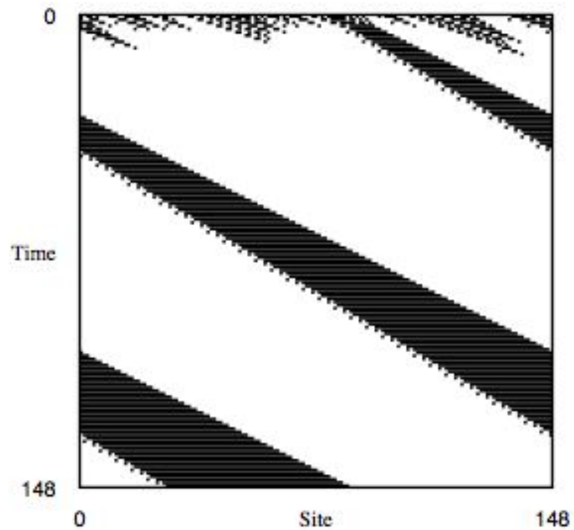



0
Time
148
0
Site
148
0
Time
148
0
Site
148

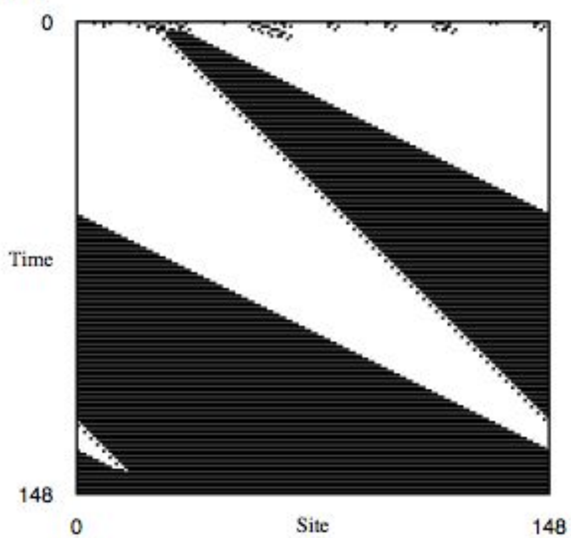
Epoch 3 Misclassifications



(a)



(b)



(c)

Figure 8: Three typical errors made by Epoch 3 rules. In (a), a rule with $\lambda \approx 0.49$ incorrectly expands blocks in an IC with $\rho_0 \approx 0.38$. In (b), a rule with $\lambda \approx 0.42$ expands blocks too slowly on an IC with $\rho_0 \approx 0.56$. In (c), a rule with $\lambda \approx 0.52$ *creates* a block that was not present in $\mathbf{s}_0$ with $\rho_0 \approx 0.19$, and expands it. All these examples led to incorrect classifications.

Does Crossover matter? YES!

	GA, xover	GA, no xover	GA, no xover, initpop 1/2	Monte Carlo
Runs reaching Epoch 3	46/50 (92%)	13/50 (26%)	22/50 (44%)	21/50 (42%)
Runs used in averages	44/50	13/50	22/50	21/50
T_2	3.6 (3.3)	65.9 (12.9)	39.3 (20.6)	44.6 (25.6)
$T_3 - T_2$	3.8 (2.8)	9.9 (5.3)	7.6 (3.5)	2.5 (2.6)

Table 2: Fraction of runs reaching Epoch 3, fraction of runs used to compute averages (for "GA, xover" case, two outlyer runs were ommitted), mean generations to onset of Epoch 2 (T_2), and mean length of Epoch 2 in generations ($T_3 - T_2$) for those runs reaching Epoch 3 by generation 99 in four different experiments. Standard deviations are given in parentheses. The experiments are: the original experiment ("GA, xover"), an experiment in which crossover was turned off ("GA, no xover"), an experiment in which crossover was turned off and the strings in the initial GA population all had $\lambda \approx 1/2$, and an experiment in which a Monte Carlo method instead of a GA was used to search the space of CA rules. The last three will be discussed in Section 11.

What is lambda?

- λ is defined the fraction of 1s in the output bits of a rule table
- For the “majority rule” rule table $\lambda = ½$
- λ_c was claimed to be the critical value of lambda near the edge of chaos.
- Mitchell earlier claimed $\lambda = ½$ because of symmetry in the density classification problem
- λ changes as the GA epochs break the symmetry of the density classification problem

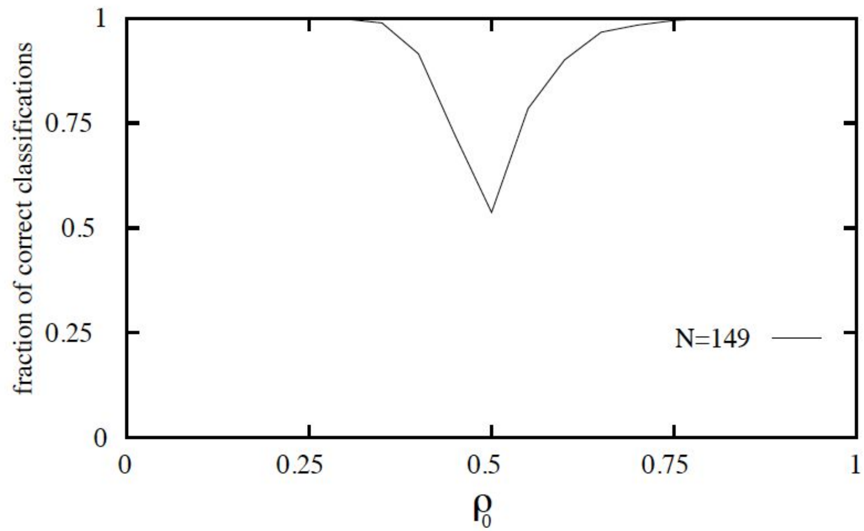
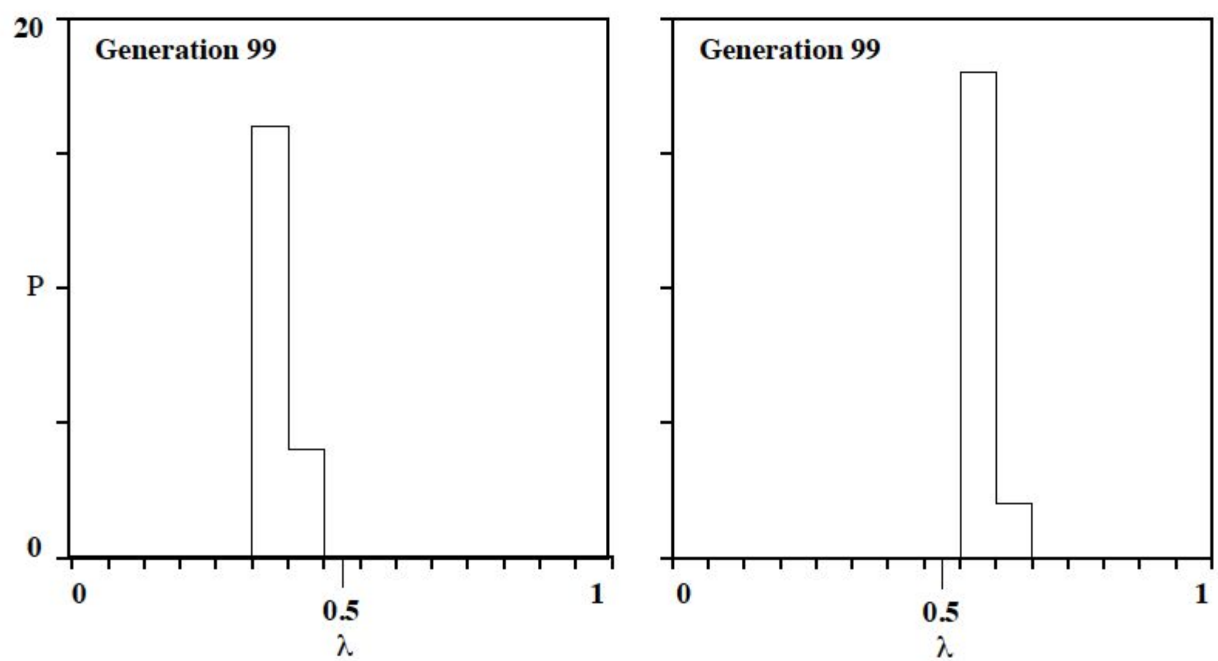


Figure 9: Performance of an Epoch 4 rule, plotted as a function of ρ_0, with $N = 149$. This rule has $\lambda \approx 0.59$.

The evolved best rules tend not occur with λ = ½



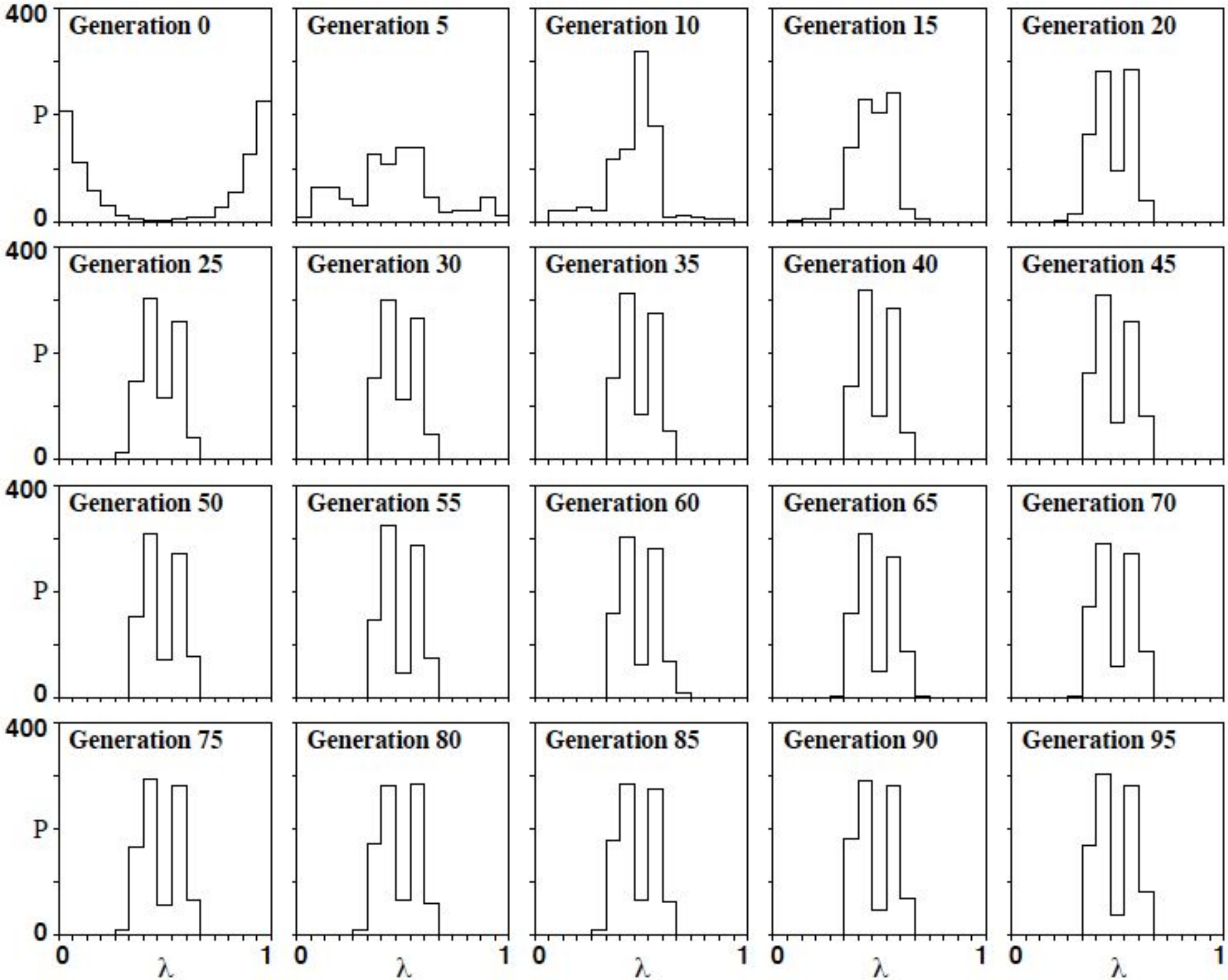


Figure 11: Frequency of elite rules versus λ given every five generations, merged from 44 GA runs with fitness function F_{100}.

Mutational Robustness

We ran the GA on populations of mutants of the GKL rule and found that many different rules have GKL-like behavior, using signals such as those described in Section 5 to classify IC density. Such rules were found at Hamming distances of up to 30 bits or more from the GKL rule. They had $F_{10^4} \approx 0.96$ and thus indicate an intermediate fitness plateau between that of the Epoch 4 rules with maximum $F_{10^4} \approx 0.945$ and that of the GKL rule with $F_{10^4} \approx 0.972$.

p35

Why did GA do well, why didn't it do better, and how could it do better?

- A_s(d) - fraction of rows in the rule table that are "majority rules"
- A_1(½) = 1 for majority rules rule (by definition all rows are majority rules
-

The temporal development of $A_s(1/2)$ and $A_s(6/7)$ for $s = 0$ and $s = 1$ averaged over the elite population helps identify how the different epochs' strategies are implemented. For rules with 7-bit neighborhoods, $|\eta| = 7$, $A_s(1/2)$ measures the degree to which $\phi(\eta)$ agrees with $\rho^s(\eta)$ in neighborhoods that have at least 4 s's ("majority agreement"). Similarly, $A_s(6/7)$ measures the degree to which $\phi(\eta)$ agrees with with $\rho^s(\eta)$ in neighborhoods that have at least 6 s's ("super-majority agreement").

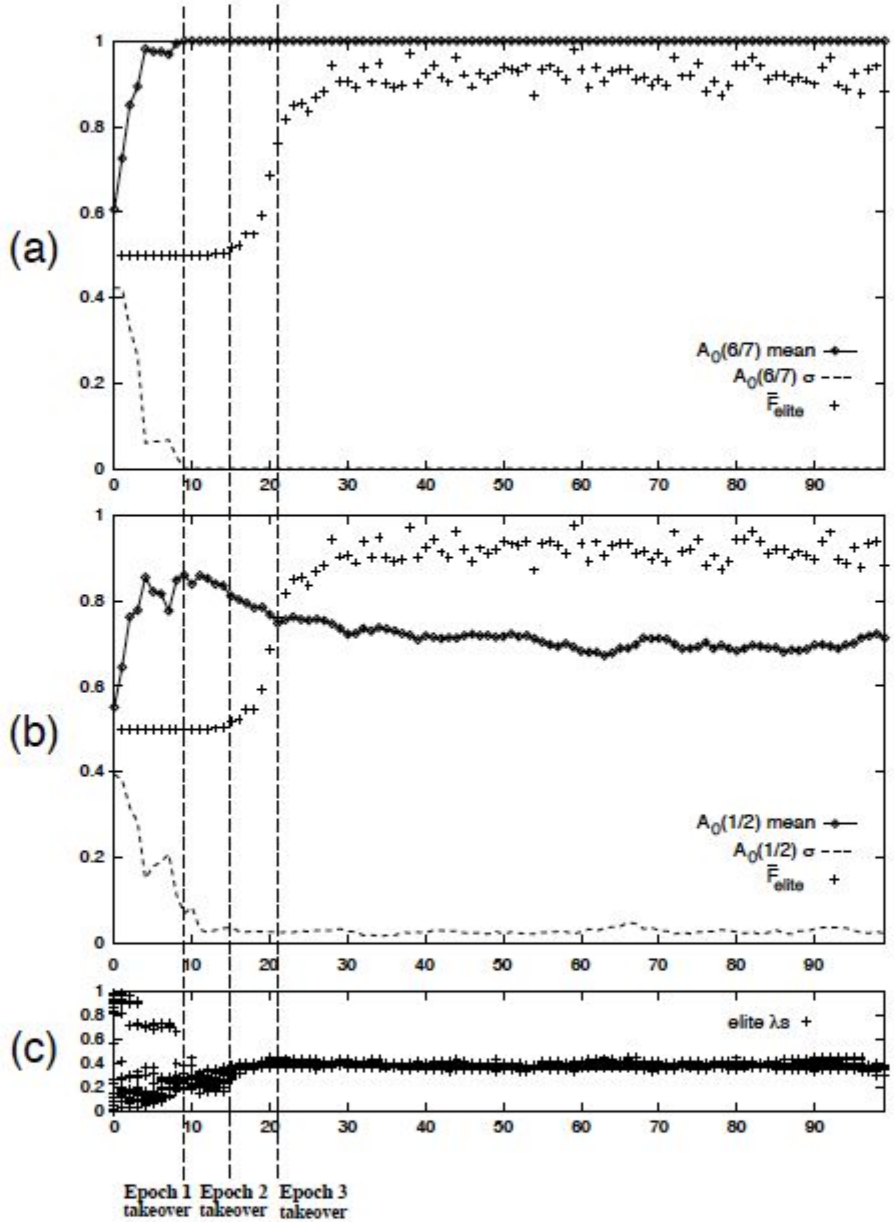


Figure 13 caption: $A_0(d)$ statistics for a run that resulted in low-ρ_0 specialists. (a) Mean and standard deviation of elite 0-agreement $A_0(6/7)$ versus generation. The mean elite fitness $\overline{F}_{\text{elite}}$ is plotted for reference. (b) Mean and standard deviation of elite 0-agreement $A_0(1/2)$ versus generation. Mean elite fitness $\overline{F}_{\text{elite}}$ is plotted for reference. (c) Scatter plot of elite λ values. The takeover generations of Epochs 1, 2, and 3 are marked by vertical dashed lines. In this run the onset generations of Epochs 1, 2, and 3—the generation in which the first instance of a new strategy was discovered (not shown here since the fitness shown is an average over the elite)—were 0, 13, and 19, respectively.

Complexity Chapter 12
Information Processing in Living Things

Information as a 3rd primitive component of reality

Living systems are information processing networks - agree?

Information processing in

- the immune system
- ant colonies
- cellular metabolism

"Computation is what a complex system does with information in order to … adapt to its environment"

Mitchell Ch 12
Information Processing in Living Systems

Information is a third primitive of reality

What plays the role of information?

Statistics and dynamic patterns among the components.

Distribution of components in time and space

How does information acquire meaning?

Meaningful communication tells the system how to respond in order to survive and reproduce (increase fitness).

…and to whom? to self aware (conscious) systems

Information Processing

Example systems

Evolution & genetic algorithms

Immune Systems

Foraging ants

How is information communicated and processed?

- Sampling - of the environment and other agents
- Randomness, balanced with determinism to achieve self regulation
- A ***parallel terraced scan*** - Fine grained **exploration** combined with focussed **exploitation**

Cellular Automata: Specifically, evolved solutions to the majority classification problem

What plays the role of information?

How is it communicated and processed

How does it acquire meaning and to whom?

Three key questions to answer about information processing

What plays the role of information?

How is it communicated and processed

How does it acquire meaning and to whom?

The University of New Mexico

Computers

What plays the role of information?
How is it communicated and processed
How does it acquire meaning and to whom?

The Immune System, Ant Colonies, Cellular Metabolism

What plays the role of information?
How is it communicated and processed
How does it acquire meaning and to whom?

Information Processing

What plays the role of information?

Statistics and dynamic patterns among the components:

How many are there and where are they?:

lymphocytes & cytokines;

ants on trails & working on specific tasks;

molecules in cells

How is it communicated and processed?

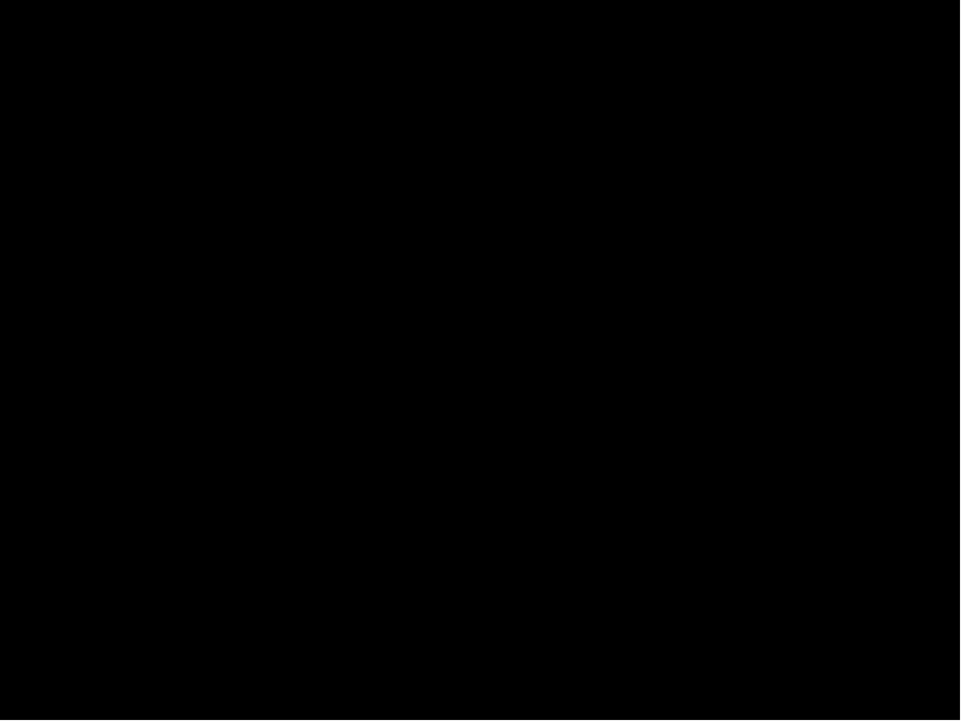
- Sampling - of the environment and other agents
- Randomness, balanced with determinism to achieve self regulation
- A ***parallel terraced scan*** - Fine grained exploration combined with focussed exploitation

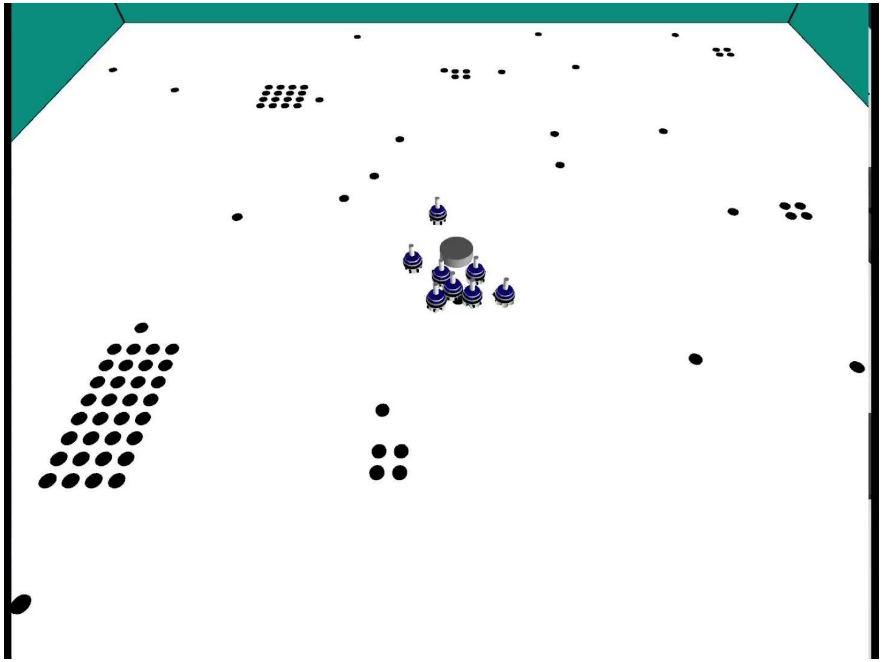
How does it acquire meaning

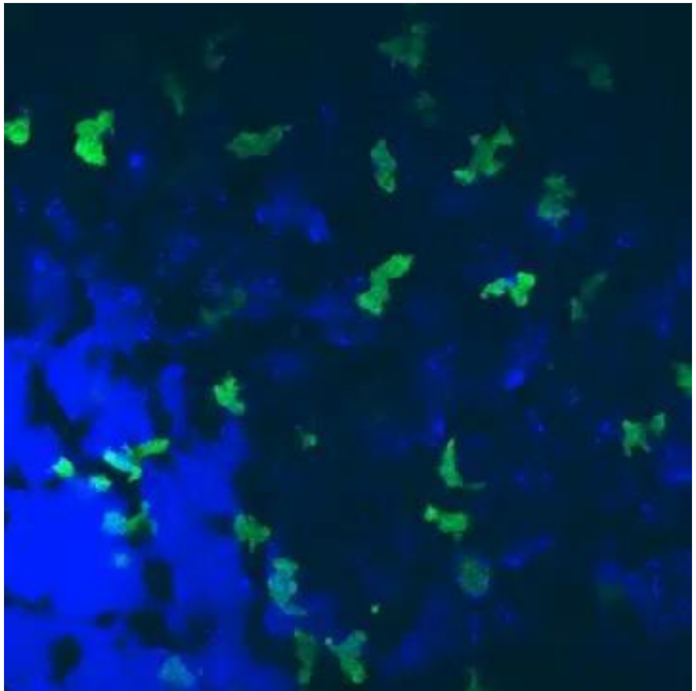
Meaningful communication tells the system how to respond in order to survive or increase fitness.

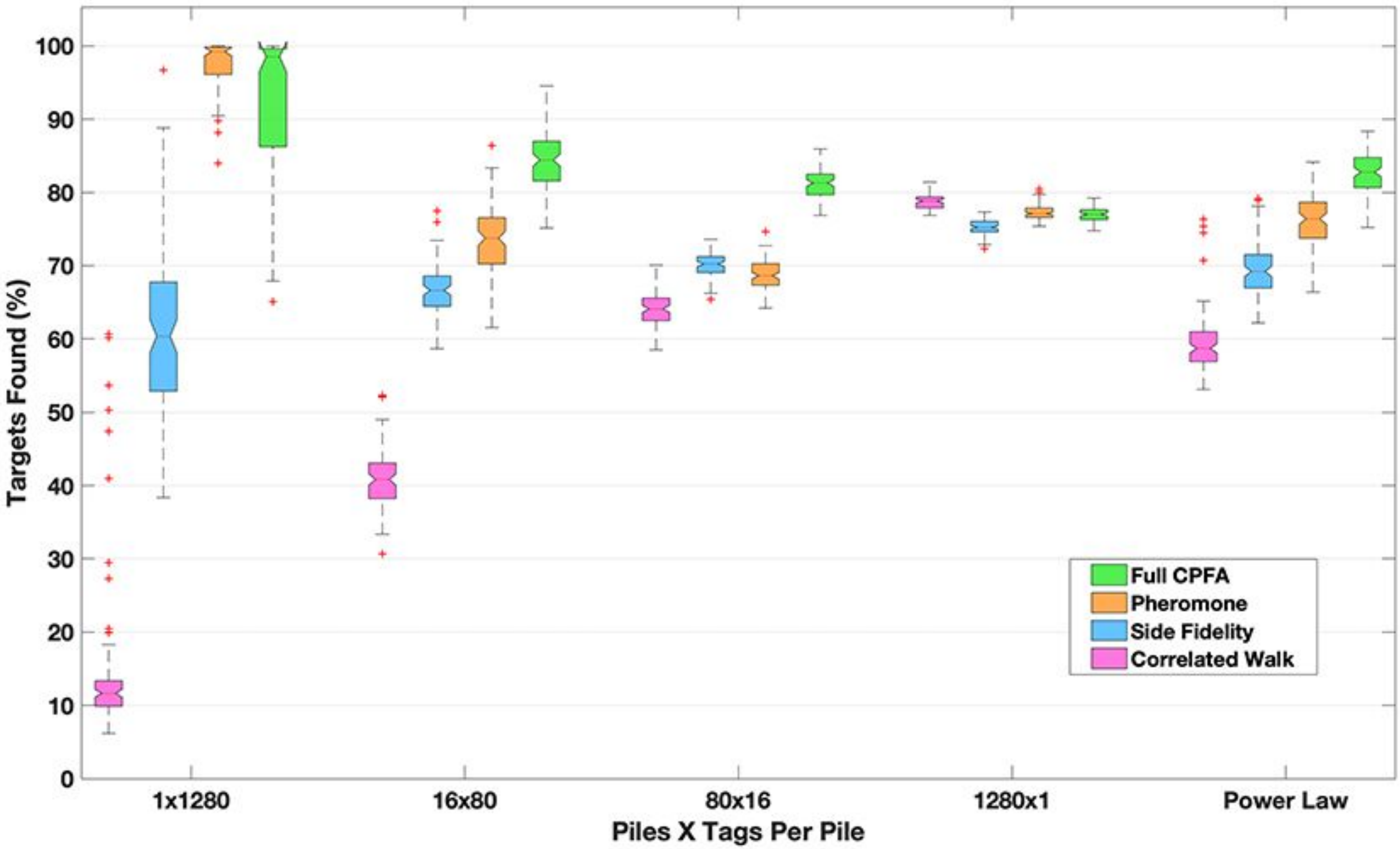
…and to whom? to self aware (conscious) systems

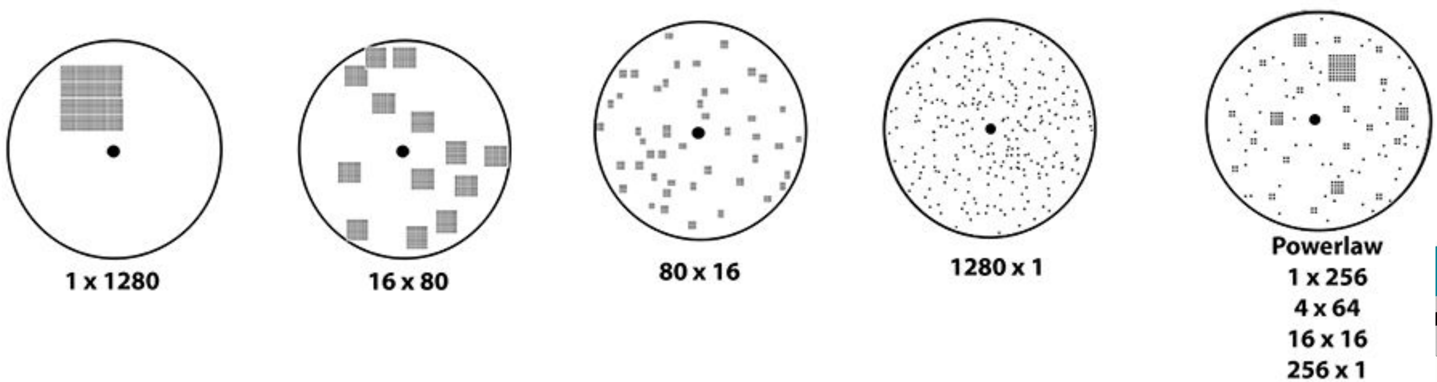
The University of New Mexico











Moses et al 2019,
https://www.frontiersin.org/articles/10.3389/fimmu.2019.01357/full