

---

# **CELLULAR AUTOMATA: A DISCRETE VIEW OF THE WORLD**

---

**Joel L. Schiff**

 **WILEY-  
INTERSCIENCE**

**A JOHN WILEY & SONS, INC., PUBLICATION**

## CHAPTER 3

---

# ONE-DIMENSIONAL CELLULAR AUTOMATA

---

*Cellular automata may be viewed as computers, in which data represented by initial configurations is processed by time evolution.*

— Stephen Wolfram – Statistical Mechanics of Cellular Automata

*The only reason for time is so that everything doesn't happen at once.*

— Albert Einstein

### 3.1 THE CELLULAR AUTOMATON

We will consider a lattice network of cells that are most commonly square in shape, but the cells can be hexagonal and other shapes as well. Each cell can exist (not simultaneously) in  $k$  different states, where  $k$  is a finite number equal to or greater than 2. One of these states has a special status and will be known as the “quiescent state.” The simplest case where each cell can exist in two possible states can

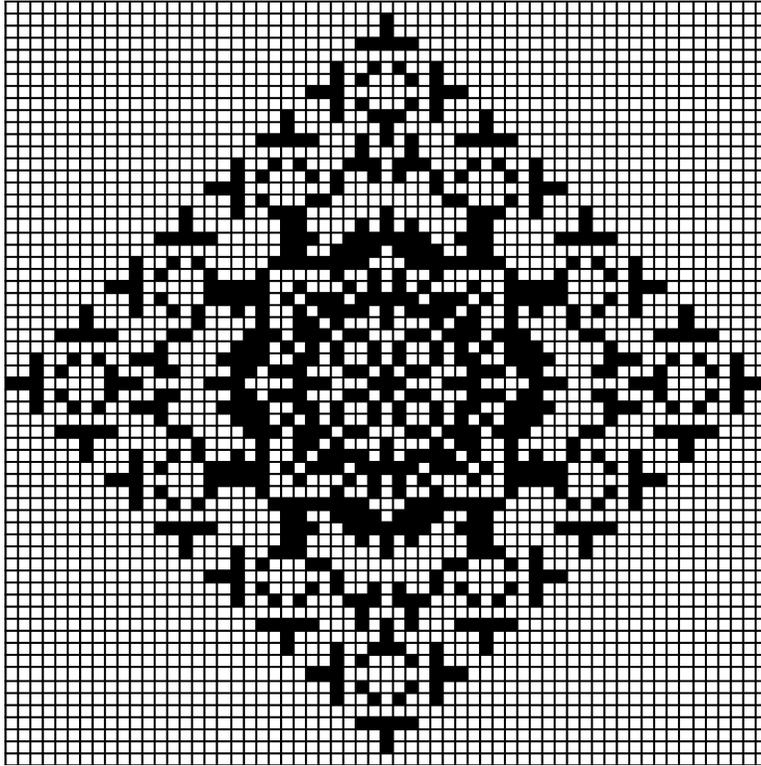
be denoted by the symbols 0 and 1 and graphically by white and black, respectively. In more anthropomorphic terms, we can think of cells in the 0 (white/quiescent) state as “dead” and those in the 1 (black) state as “alive” (see Fig. 3.1).

The lattice of cells can be  $n(\geq 1)$  dimensional, but most of the work on cellular automata (CA) has been for one and two dimensions, and we will have no recourse for any higher dimensions. In the sequel, we shall generally consider square cells as the basic unit of our automata. In the one-dimensional case, these form a row of adjacent boxes. In principle, the number of boxes in any array is infinite, but for practical purposes it will simply be taken sufficiently large to illustrate the behavior in question, often with certain conditions imposed on the cells along the boundaries. In some cases, intentional restrictions will be made on the size of the lattice as well.

In order for the cells of our lattice to evolve, we need time to change. This is done in an unusual manner by considering changes to the states of cells in the lattice only at discrete moments in time, that is, at *time steps*  $t = 0, 1, 2, 3, \dots$ , as in the ticking of a clock. The time  $t = 0$  is commonly reserved for the initial time period before any change of the cells’ states has taken place. One further ingredient that is needed for our cellular lattice to evolve with discrete time steps is a local rule or *local transition function* governing how each cell alters its state from the present instant of time to the next based on a set of rules that take into account the cell’s current state and the current state of its neighbors. Of course, which cells are considered to be neighbors needs to be precisely defined. This alteration of cell states takes place *synchronously* for all cells in the lattice. The transition function can be deterministic or probabilistic, but in most cases (some exceptions are models discussed in Chapter 5 on applications) we will consider only the former. The lattice of cells, the set of allowable states, together with the transition function is called a *cellular automaton*. Any assignment of state values to the entire lattice of cells by the transition function results in a *configuration* at any particular time step.

Therefore the three fundamental features of a cellular automaton are:

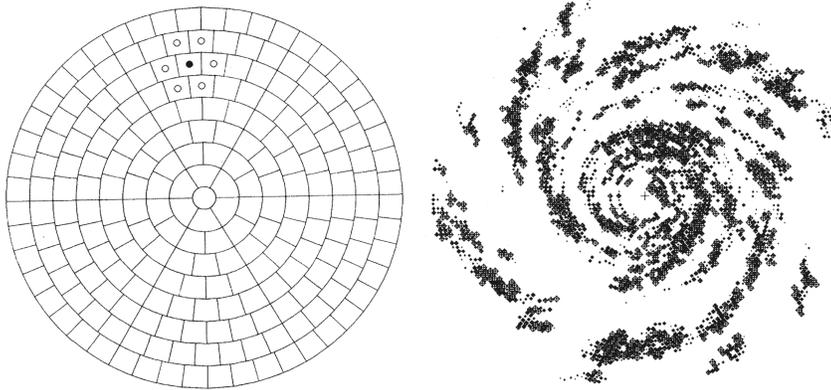
- *uniformity*: all cell states are updated by the same set of rules;
- *synchronicity*: all cell states are updated simultaneously;
- *locality*: the rules are local in nature.



**Figure 3.1** A two-dimensional configuration with each cell taking one of two state values, black or white according to a particular local transition function. The first printed reference to a system such as the above, even using the word *automata*, is to be found in the paper by Ulam [1950].

There has been some recent work on the *asynchronous* updating of cell states (cf., e.g., Bersini and Detour [1994], Ingerson and Buvel [1984], Lumer and Nicolis [1994], Schönfisch and de Roos [1999]). This can be achieved in a number of different ways and is discussed in Section 4.4.

Other liberties with the rules of a cellular automaton can be taken. In one interesting example, galaxy formation has been simulated by Shulman and Seiden [1986] using a CA approach on a polar grid whose rings rotate at different rates (see Fig. 3.2). Theirs was a simple percolation model and achieves the basic structure of a spiral galaxy without a detailed analysis of the astrophysical dynamics involved.



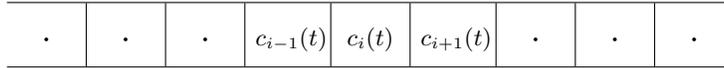
**Figure 3.2** In this setting the neighbors of each cell change due to the differential rotation of the rings of the polar grid that is used to emulate galaxy formation. The black circle is an active region of star formation which induces star formation in its neighbors with a certain probability at the next time step. At right is a typical galaxy simulation.

In another digression from the conventional CA definition, Moshe Sipper [1994] considered different rules for different cells as well as the evolution of rules over time. There will be other instances in the sequel when we will stray slightly from strict adherence to the classical cellular automaton definition.

A fundamental precept of cellular automata is that the local transition function determining the state of each individual cell at a particular time step should be based upon the state of those cells in its immediate neighborhood at the previous time step or even previous time steps. Thus the rules are strictly local in nature and each cell becomes an information processing unit integrating the state of the cells around it and altering its own state in unison with all the others at the next time step in accordance with the stipulated rule. Thus, many global patterns of the system are an *emergent* feature of the effect of the locally defined transition function. That is, complex global features can emerge from the strictly local interaction of individual cells each of which is only aware of its immediate environment. This emergent behavior will be discussed more comprehensively in Chapter 6, but it is a salient feature that one should always be aware of in our development of cellular automata theory.



**Figure 3.3** The eight possible neighborhood-states with  $r = 1$  and  $k = 2$ . Here, as will be our usual convention, black is 1 and white is 0.



**Figure 3.4** The cell states of the central cell  $c_i$  and its two nearest neighbors  $c_{i-1}$  and  $c_{i+1}$  at the time step  $t$ .

### 3.2 TRANSITION FUNCTIONS

Most of the dynamical features of cellular automata can be found in the study of the one-dimensional case. Here we define a neighborhood of a cell  $c$  having radius (range)  $r$  as the  $r$  cells to the left of  $c$  and the same number of cells to the right of  $c$ . Counting  $c$  itself, this neighborhood contains  $2r + 1$  cells. In the simplest case  $r = 1$  and  $k = 2$  for the allowable states. In this instance, a three-cell neighborhood with two different states 0 and 1 for each cell can be expressed in  $2^3 = 8$  different ways. All eight neighborhood-states with  $r = 1$  and  $k = 2$  are illustrated below (Fig. 3.3), and in general there are  $k^{2r+1}$  one-dimensional neighborhood-states.

Let us adopt this notation:  $c_i(t)$  denotes the state of the  $i$ th cell at time  $t$  (Fig. 3.4).

At the next time step,  $t + 1$ , the cell state will be  $c_i(t + 1)$ . Mathematically we can express the dependence of a cell's state at time step  $t + 1$  on the state of its left-hand and right-hand nearest neighbors  $c_{i-1}(t)$  and  $c_{i+1}(t)$  at time step  $t$  by the relation

$$c_i(t + 1) = \varphi [c_{i-1}(t), c_i(t), c_{i+1}(t)],$$

where  $\varphi$  is the local transition function. For example, consider the simple transition rule

$$c_i(t + 1) = c_{i-1}(t) + c_i(t) + c_{i+1}(t) \pmod{2}, \quad (3.1)$$

where mod 2 means taking the remainder after division of the indicated sum by 2, resulting in either 0 or 1. We can put this rule into a *transition table* format by adding  $c_{i-1}(t) + c_i(t) + c_{i+1}(t) \pmod{2}$  for the eight possible different input values:

$c_{i-1}(t)$	$c_i(t)$	$c_{i+1}(t)$	$c_i(t+1)$
1	1	1	1
1	1	0	0
1	0	1	0
1	0	0	1
0	1	1	0
0	1	0	1
0	0	1	1
0	0	0	0

(3.2)

A very convenient way to illustrate the allowable rules for one-dimensional cellular automata with  $r = 1$  and  $k = 2$  is to indicate the state (color) of the middle cell at the next time step, given the state (color) of itself and its two nearest neighbors (Fig. 3.5).

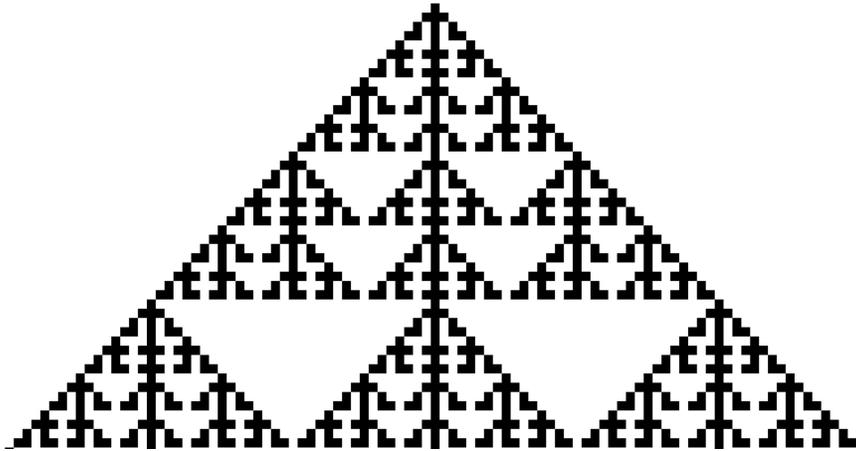
Here the middle cell at time  $t$  (in the top row) has its state altered according to the state of its two nearest neighbors and its own state to yield the cell's new state at time  $t + 1$  (bottom row). This also represents the rule in Equation 3.1. Observe that in four of the eight cases a black cell appears. One consequence of this is that if we took a disordered array of a large number of cell sites, then the average fraction (or *density*) of black cell sites that evolve after one iteration will be 0.5.

The cellular automaton defined by this rule has the graphical representation depicted in Fig. 3.6 starting with a single black cell with each subsequent generation of the automaton appearing on the next line down.

So how many possible rules are there for  $r = 1$  and  $k = 2$ ? Since there are eight possible neighborhood-states of three cells and each of these results in two possible state outcomes for the middle cell, there are  $2^8 = 256$  possible transition rules by which this can be achieved. Wolfram described these as *elementary* cellular automata. We can also identify the rule given in the form of the transition table 3.2, by its eight output states (or *rulestring*) 10010110, which in base 2 happens to be the number 150. So this rule is referred to as Rule 150 and is the rule



Figure 3.5



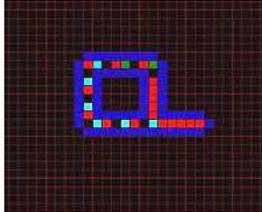
**Figure 3.6** By considering a white cell as being in state 0 and a black cell in state 1, we can depict the evolution of the rule in the text starting with a single black cell. Each new line downward represents the evolution of the automaton at the next time step.

above depicted now in five different formats: Equation 3.1, transition table 3.2, the rulestring, and Figs. 3.5 and 3.6.

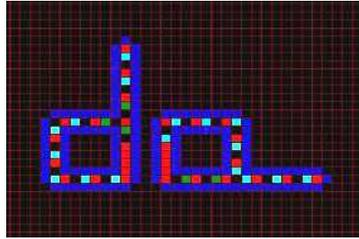
In a similar fashion, all 256 elementary cellular automata can be numbered starting from Figs. 3.7 to 3.10.

A complete illustration of all 256 elementary cellular automata starting with a standard initial condition of one black cell is given in Appendix A. Many of the salient features found in cellular automata theory can be observed in these elementary ones and we shall refer to them often. These elementary cellular automata are examples of *first order automata* in the sense that the state of a cell at time step  $t + 1$  only depends on the state of its neighbors at the previous time step  $t$ . Whereas in a *second order automaton* a cell's state at time step  $t + 1$  is rather more demanding and depends on the state of its neighbors at time steps  $t - 1$  as well as  $t$ , analogous to the way the Fibonacci sequence was formed. Unless otherwise specified, we will mainly be considering first order automata, although second order automata will come into play in our discussion of reversibility.

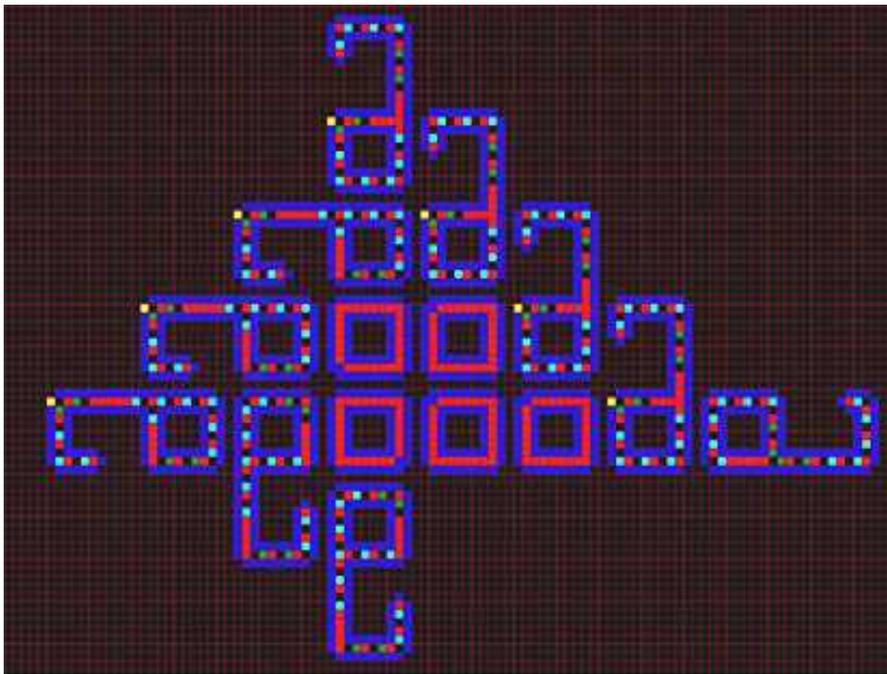
Perhaps you are thinking that this is not so much to work with, so if we merely increase the radius to  $r = 2$ , then there are  $2^5 = 32$  possible neighborhood-states and  $2^{32} = 4,294,967,296$  possible rules. On the



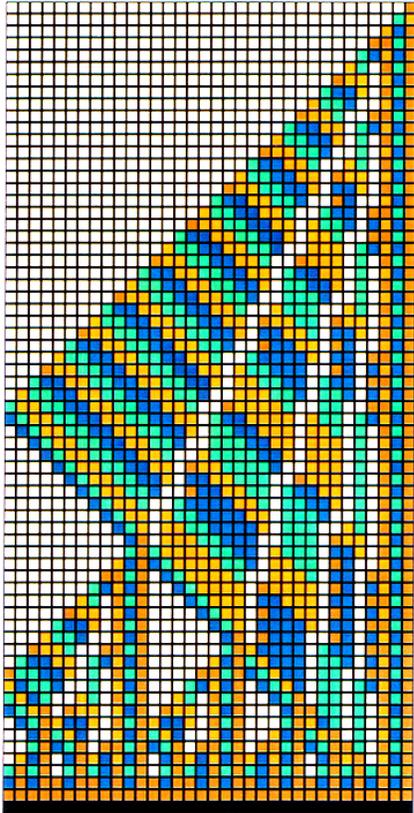
**Figure C.1** The Langton looped pathway in its initial configuration. The different colors represent the different cell states.



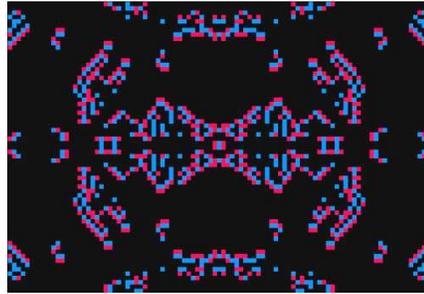
**Figure C.2** The parent (left) and offspring loops each propagating new construction arms after their connection has been severed.



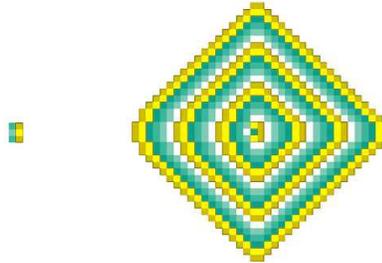
**Figure C.3** The replication continues indefinitely filling up the plane with loops.



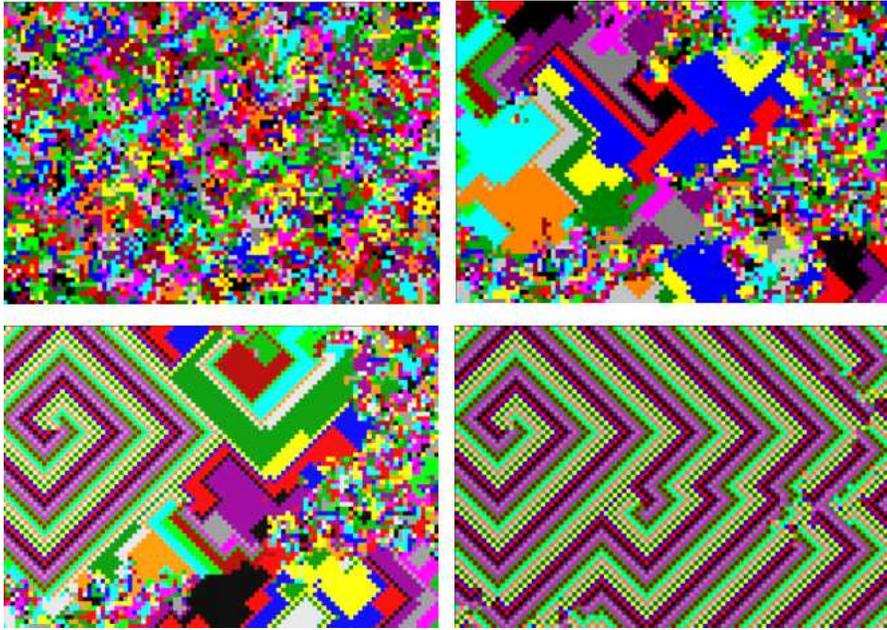
**Figure C.4** The minimal time solution of Jacques Mazoyer of the firing squad synchronization problem using 6 states. Here  $n = 34$  and time increases from top to bottom. The general is on the right-hand side and also reaches the firing state (black) together with all the soldiers in 67 time steps.



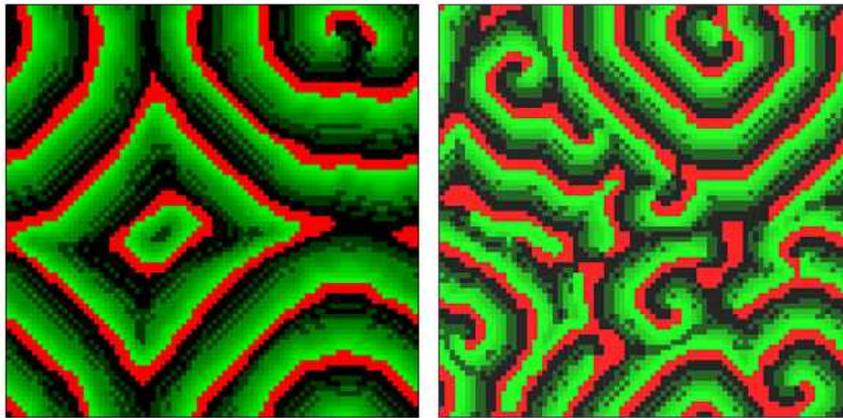
**Figure C.5** The three-state cellular automaton Brian's Brain created by Brian Silverman. Waves of live cells tend to sweep across the array. Here the colors are: black = ready, red = firing, blue = refractory.



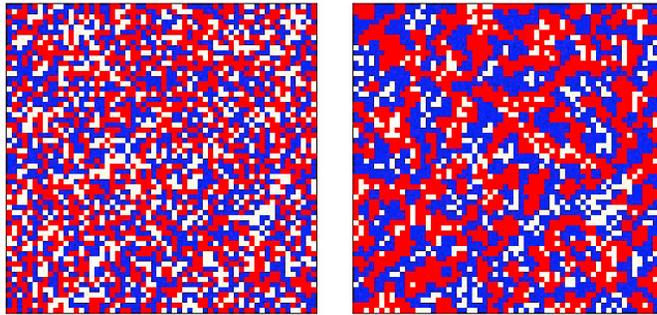
**Figure C.6** In this Greenberg-Hasting model using the Moore neighborhood, there are 2 excited stages (yellow) and 3 refractory stages (green). As above the transition at each stage is from darkest to lightest. White is the rested state. At least 2 excited neighbors are required for a cell in the rested state to become excited. The figure shows the initial configuration (left) and after 19 iterations (right). Waves are generated in the center of the figure and grow and move outward.



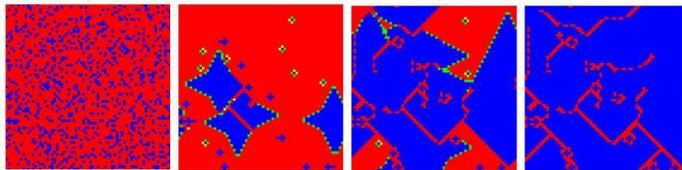
**Figure C.7** The time evolution from a random initial state to the formation of waves of the cyclic space automaton. The frames (left to right) represent the debris, droplet, defect, and demon phases, respectively.



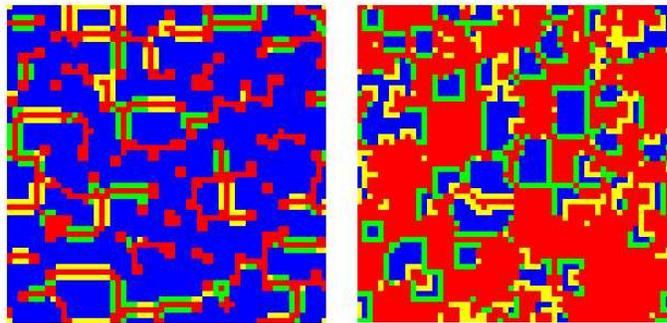
**Figure C.8** Output from the hodgepodge machine using  $g = 30$  (left) and  $g = 40$  (right), with the other parameters given in the text. Notice the wave and spiral formations.



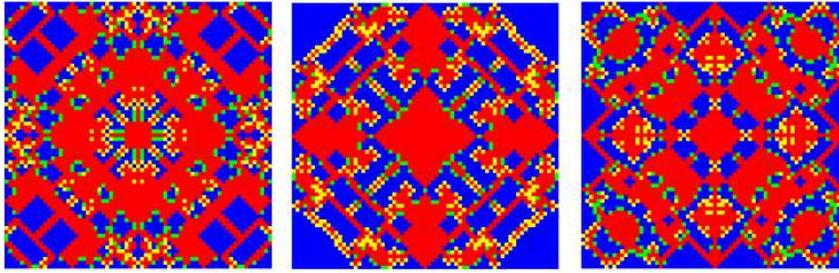
**Figure C.9** In this Schelling model implementation, the left-hand side is an initial random distribution of occupied cell sites (25% white vacant space) and the right-hand side represents the segregated equilibrium state of the red and blue cells reached after applying the contentment requirement that 37.5% of the neighbors should be of the same kind.



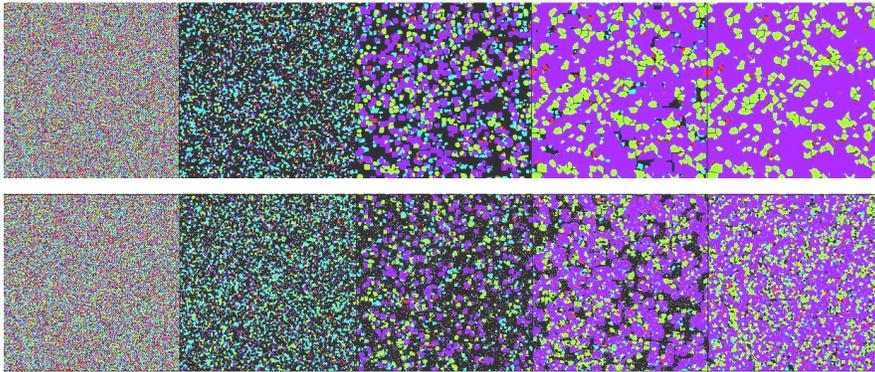
**Figure C.10** The evolution of cooperation (blue) in the spatial prisoner's dilemma, starting with a random array of 80% defectors (red) using a von Neumann neighborhood. The payoffs are:  $\{R; S; T; P\} = \{3; 0; 3.5; 0.5\}$ .



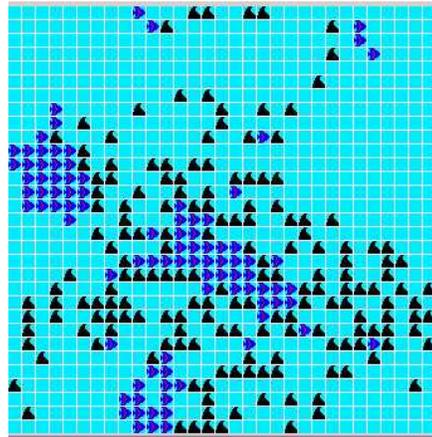
**Figure C.11** Using the Nowak and May payoffs with  $b = 1.6$  (left) and  $b = 1.85$  (right), with an initial distribution of 10% defectors distributed randomly. Here the central cell also plays against itself using a nine-cell neighborhood. Color code is: blue = cooperators, green = new cooperators, red = defectors, yellow = new defectors. Periodic boundary conditions are used. Cooperation is seen to decrease with increasing values of the temptation parameter  $b$ .



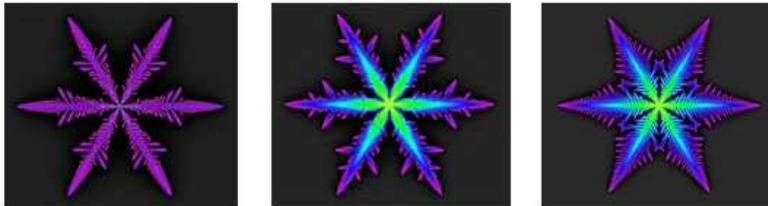
**Figure C.12** Employing a von Neumann neighborhood, these are three images in the time evolution of the spatial prisoner's dilemma with a value of  $b = 1.4$  and an initial sole defector set in a sea of cooperators. Periodic boundary conditions are used. The color code is: blue = cooperator, green = new cooperator, red = defector, yellow = new defector.



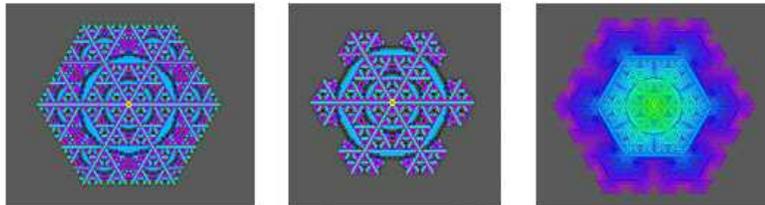
**Figure C.13** The evolution of the spatial iterated prisoner's dilemma with five competing strategies with the standard payoffs from an initial random configuration. Each cell's strategy plays off five times against its eight neighbors and then adopts the strategy of its most successful neighbor. Color code: Always Cooperate = red, Tit-for-Tat = purple, Random = cyan, Pavlov = green, and Always Defect = black. The predominance of Tit-for-Tat over time is evident, with the Random strategy hardly noticeable. The second row has a certain level of noise introduced, that is, a cell will make a random move in a single round, thus increasing the dynamics of the evolution. Pavlov now holds its own against Tit-for-Tat in this instance. Images are taken at the initial state, after 2, 5, 10, and 50 iterations.



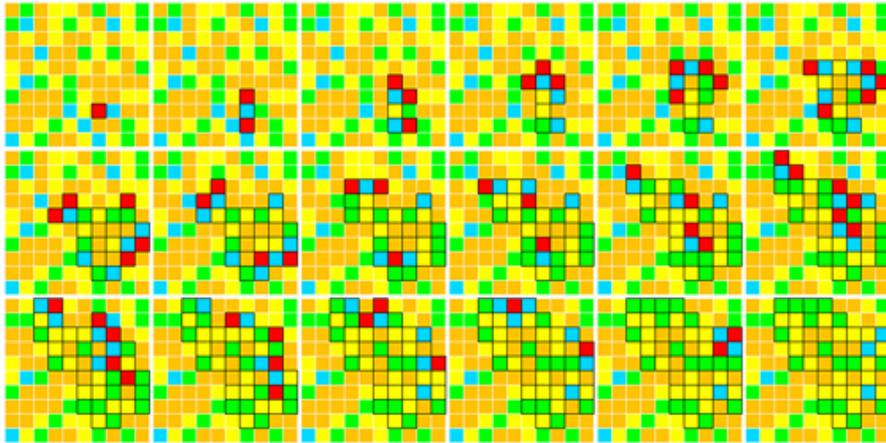
**Figure C.14** The WATOR-World of fish (dark blue), sharks (black), and water (pale blue). There is an initial random mixing of 200 fish and 5 sharks. The image is taken at the 297th time step where the fish population is now 93 and the shark population has risen to 177. Breeding ages for fish and sharks is 2 and 10 generations, respectively, and the starvation time for sharks has been set at 3. The geometry of the local population mix can affect the dynamics.



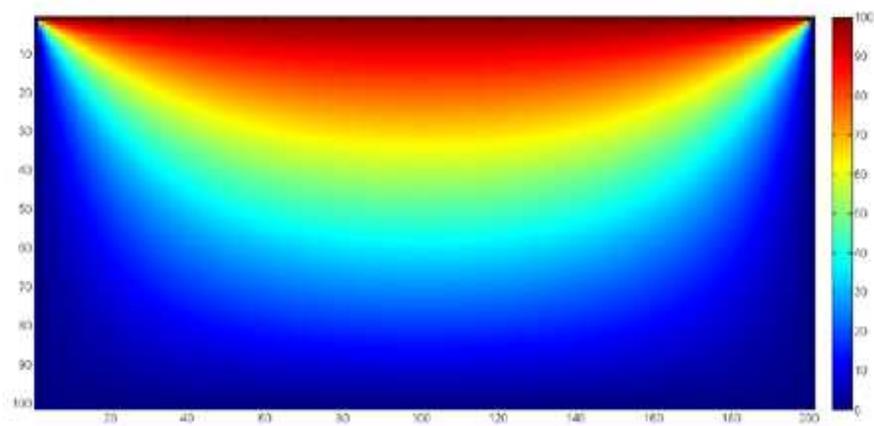
**Figure C.15** Stellar dendrite snow crystal forms with  $\beta = 0.35, \gamma = 0$ ;  $\beta = 0.4, \gamma = 0.001$ ;  $\beta = 0.45, \gamma = 0.001$ , respectively.



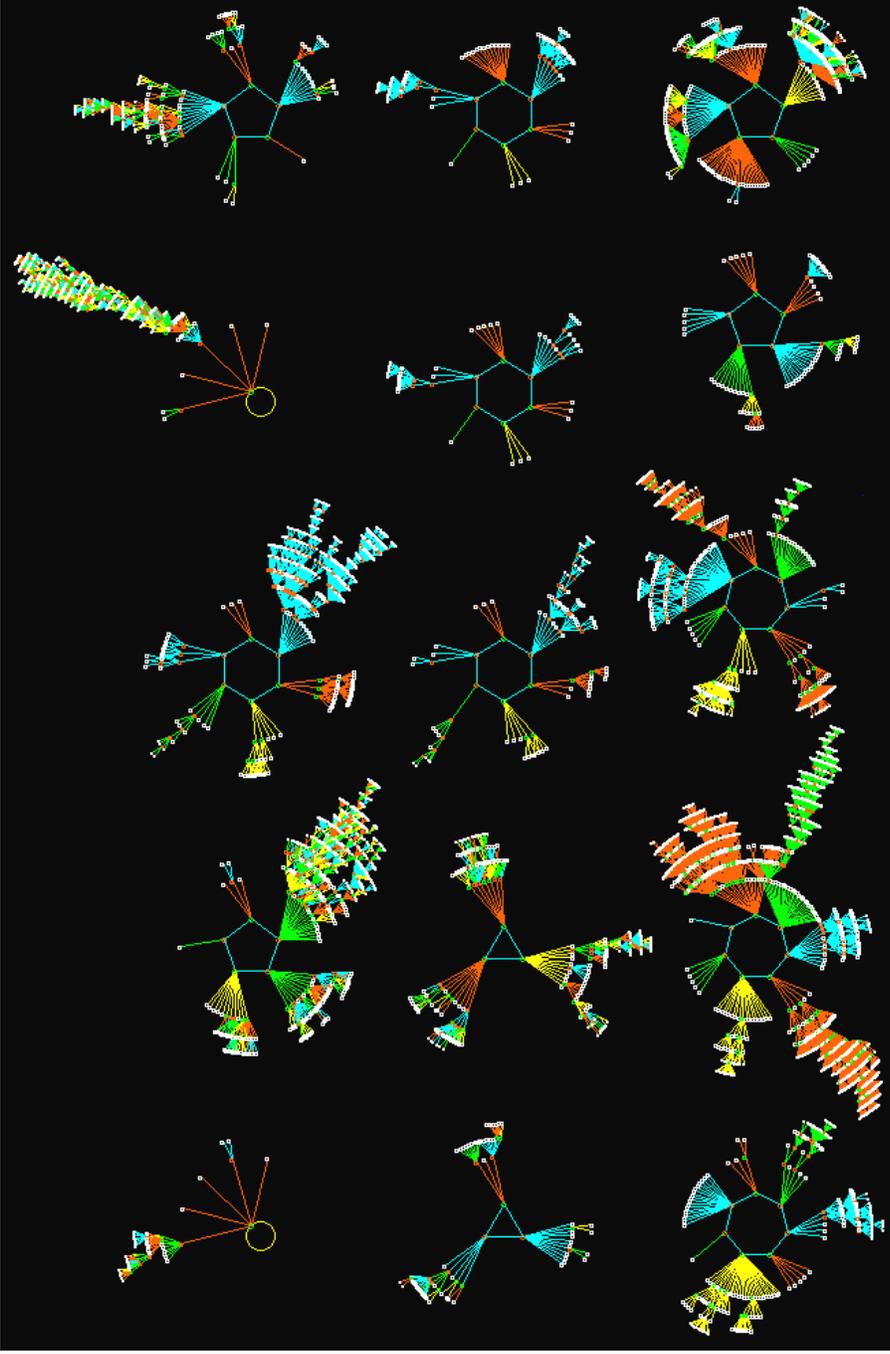
**Figure C.16** Plate form snow crystals. The first two have the same parameter values,  $\beta = 0.95, \gamma = 0$ , and represent different time steps. Third image has  $\beta = 0.95, \gamma = 0.0035$ .



**Figure C.17** The time evolution of a single avalanche (cascade of topplings) precipitated by the red cell in the first frame until a stable state is reached in the last frame. Here blue = 0, green = 1, yellow = 2, orange = 3, red = 4 and the topplings start when the site value is at least 4. The boundary was held at state 0 but is omitted from the figures.



**Figure C.18** Here we have color coded the steady-state temperature values of our cellular automaton simulation from red (hot) to blue (cold). See text for details.



**Figure C.19** These are not galaxies photographed by the Hubble Space Telescope. They are basins of attraction of an RBN with  $N = 13$ ,  $K = 3$ . The state space consisting of  $2^{13} = 8192$  states is organized into 15 basins having attractor periods ranging from 1 to 7.