

## 2. Cellular Automata

“Cellular automata are sufficient simple to allow detailed mathematical analysis, yet sufficient complex to exhibit a wide variety of complicated phenomena.”

Wolfram (1983)

*Cellular automata (CA) cannot and should not be covered comprehensively in this book. The current chapter shall give the reader a glimpse of the manifold arrangements and the peculiarities of CA. It will serve as a background for the discussion of a special type of cellular automata, namely lattice-gas cellular automata.*

### 2.1 What are cellular automata?

CA can be characterized as follows (e.g., Wolfram, 1984b or Hedrich, 1990; see below for a formal definition):

- CA are regular arrangements of single *cells* of the same kind.
- Each cell holds a finite number of discrete states.
- The states are updated simultaneously (‘synchronously’) at discrete time levels.
- The update rules are deterministic and uniform in space and time.
- The rules for the evolution of a cell depend only on a local neighborhood of cells around it.

Not all of these criteria are always fulfilled. The cells can be positioned, for example, at the nodes of a (quasiperiodic) *Penrose lattice* (Penrose, 1974, 1979) or at random (Markus and Hess, 1990). A random connection of cells was proposed by Richard Feynman (Hillis, 1989). The update rules of certain CA include probabilistic elements (compare the FHP lattice gas automata, Section 3.2).

The formal definition of CA follows Kutrib et al. (1997). The cells can be imagined as positioned at the integer points of the  $D$ -dimensional Euclidean lattice  $\mathcal{L} = \mathbb{Z}^D$ . The *finite set of possible states* of each of the cells is equal and will be denoted by  $Q^1$ .

The state of a cell  $i$  at a new time level  $t + 1$  depends on the states of cells  $j$  in a finite neighborhood<sup>2</sup>  $N \subset \mathbb{Z}^D$  at time  $t^3$ . The elements  $n \in N$  are to be interpreted as the *relative coordinates* of neighboring cells (with  $(0, \dots, 0)$  as relative coordinate of cell  $i$ ). The neighborhood  $N$  may be interpreted as an interconnection between the cells.

A mapping  $l : N \rightarrow Q$  is called a *local configuration*<sup>4</sup>. It contains exactly the information to update a cell. The mode of operation of a cell is completely determined by its *local rule*  $r : Q^N \rightarrow Q$  where  $Q^N$  is the set of all mappings  $f : N \rightarrow Q$ . The CA updating is called homogeneous when the neighborhoods  $N$  and  $N'$  of the cells  $i$  and  $i'$  map onto each other by a translation and when the same local rule is applied to all cells.

The *global configuration* of a CA (i.e. the ensemble of the state of all cells) at a certain time is called a (*global*) *configuration*  $g$ . CA are working in discrete time. The global configuration  $g$  at time  $t$  leads to a new global configuration  $g'$  at time  $t + 1$  whereby all cells enter a new state according to the local rule *synchronously*. The associated *global rule* is a mapping  $R : Q^{\mathcal{L}} \rightarrow Q^{\mathcal{L}}$ .

## 2.2 A short history of cellular automata

Around 1950 cellular automata<sup>5</sup> were introduced by Stanislas Ulam, John von Neumann, and Konrad Zuse<sup>6</sup>. Ulam simulated the growth of patterns in two and three dimensions (compare Ulam, 1952 and 1962; Schrandt and Ulam, 1970). John von Neumann proposed a self-reproducing cellular automaton (von Neumann, 1966) which at the same time realized a universal *Turing machine* (Turing, 1936; Hopcroft, 1984). Each of the approximately 200000 cells of von Neumann's CA holds 29 different states. A few years ago this CA has been implemented for the first time on a computer (Signorini, 1989).

<sup>1</sup> For example,  $Q = \{0, 1\}$  for a binary automaton

<sup>2</sup> The neighborhood includes the cell  $i$ . There are, however, particular update rules that do not depend on the state of  $i$  at time  $t$ .

<sup>3</sup> A random process whose future probabilities are determined by its most recent values is called a Markov process. If not otherwise stated the updating of CA is, however, a deterministic process.

<sup>4</sup> We will also denote an actual state of a neighborhood  $N$  as the local configuration.

<sup>5</sup> Other names: cellular spaces, tessellation automata, homogeneous structures, cellular structures, tessellation structures, iterative arrays.

<sup>6</sup> Some scientists even regard the paper by Wiener and Rosenblueth (1946) as the first one in this field.

Zuse published his ideas concerning the application of cellular automata to physical problems in a monograph (Zuse, 1969; English translation 1970). Some of his formulations already resemble to the HPP lattice-gas cellular automata proposed four years later by Hardy et al. (1973). In addition to hydrodynamic problems Zuse had in mind models for electrodynamics and quantum theory. The most far-reaching vision was his concept of the universe as a cellular automaton encompassing a gigantic number of cells (Zuse, 1982).

As far as number of citations can tell something about the flow of ideas, Zuse's monograph (1969; 1970) did not have a major impact (but see Alasyev et al., 1989; Case et al., 1990; Fredkin, 1990; Toffoli, T. and N. Margolus, 1990; Rothman and Zaleski, 1994).

In 1970 John Horton Conway introduced the game '*Life*', a two-dimensional CA with simple update rules but complex dynamics (compare Section 2.4.3). Martin Gardner made cellular automata very popular by a series of papers on '*Life*' in Scientific American (Gardner, 1970, 1971a,b,c; see also: Berlekamp, Conway and Guy, 1984).

The first lattice-gas cellular automata (LGCA) - special kinds of cellular automata for the simulation of fluid flow and other physical problems - was proposed in 1973 by Hardy, Pomeau and de Pazzis. Its name HPP is derived from the initials of the three authors. Although the HPP model conserves mass and momentum it does not yield the desired Navier-Stokes equation in the macroscopic limit.

In 1983 Stephen Wolfram revived the interest in CA by a series of papers (Wolfram, 1983, 1984a,b,c). The one-dimensional arrays of cells considered by Wolfram expressed complex patterns when initialized randomly and updated by simple deterministic rules depending on the state of the cell and a few of its neighbors.

In 1986 Frisch, Hasslacher and Pomeau discovered that a CA over a lattice with hexagonal symmetry, i.e. with a somewhat higher symmetry than for the HPP model, leads to the Navier-Stokes equation in the macroscopic limit. The theoretical foundations of lattice gas automata were given soon after by Wolfram (1986) and Frisch et al. (1987).

## 2.3 One-dimensional cellular automata

*Wolfram (1983, 1984a,b) investigated one-dimensional cellular automata. He introduced a division into four universal classes. Even the study of the branch concerned with one-dimensional cellular automata is far from completed because only a small subset of possible rules has been explored and a theoretical understanding is still in its infancy (Wolfram, 1985).*

One-dimensional cellular automata consist of a number of uniform cells arranged like beads on a string. If not stated otherwise arrays with finite number of cells and periodic boundary conditions will be investigated, i.e. the beads form a necklace (compare Fig. 2.3.1). The states of all cells form a (*global*) *configuration* of a CA.

**Fig. 2.3.1.** *One-dimensional cellular automata with the two possible states per cell: empty or occupied (marked with a cross).*



The state of cell  $i$  at time  $t$  is referred to as  $a_i^{(t)}$ . The finite number of possible states  $k < \infty$  are labelled by non-negative integers from 0 to  $k - 1$ , i.e.  $a_i^{(t)} \in \mathbb{Z}_k$  (mathematicians call the set of integers modulo  $k$  the residue class  $\mathbb{Z}_k$ ). The state of each cell develops in time by iteration of the map

$$a_i^{(t)} = F[a_{i-r}^{(t-1)}, a_{i-r+1}^{(t-1)}, \dots, a_i^{(t-1)}, \dots, a_{i+r}^{(t-1)}] \quad (2.3.1)$$

i.e. the state of the  $i$ th cell at the new time level  $t$  depends only on the state of the  $i$ th cell and the  $r$  (range) neighbors to the left and right at the previous time level  $t - 1$ . The arbitrary (in general nonlinear) function  $F$  is called the *automata rule*, the *update rule* or just the *rule*. An alternative formulation of the rule (2.3.1) reads

$$a_i^{(t)} = f \left[ \sum_{j=-r}^{j=r} \alpha_j a_{i+j}^{(t-1)} \right] \quad (2.3.2)$$

where the  $\alpha_j$  are integer constants and thus the function  $f$  has a single integer as argument.

**Exercise 2.3.1.** (\*\*)

Why can (2.3.1) and (2.3.2) be equivalent formulations?

**Number of automata rules.** Consider a CA with  $k = 2$  possible states per cell and a range  $r = 1$ . The possible combinations of the arguments of the automata rule  $F$  are listed in two different representations in Tables 2.3.1 and 2.3.2.

There are 8 different combinations (in general:  $k^{2r+1}$ ). Interpretation of  $\{a_{i-1}^{(t-1)}, a_i^{(t-1)}, a_{i+1}^{(t-1)}\}$  (columns 2 to 4 in Table 2.3.1) as the bit pattern (with the highest bit to the left) of an integer in binary representation yields the

**Table 2.3.1.** *An example of an automata rule for a CA with  $k = 2$  and  $r = 1$ .*

	$a_{i-1}^{(t-1)}$	$a_i^{(t-1)}$	$a_{i+1}^{(t-1)}$	$a_i^{(t)}$
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	0
7	1	1	1	0

**Table 2.3.2.** *An example of an automata rule for a CA with  $k = 2$  and  $r = 1$ .*

<u>111</u>	<u>110</u>	<u>101</u>	<u>100</u>	<u>011</u>	<u>010</u>	<u>001</u>	<u>000</u>
0	0	0	1	0	0	1	0

numbers 0 to 7 (listed in the first column). In the last column of Table 2.3.1 one of the possible rules is given in tabular form. It consists of a certain sequence of zeros and ones which also can be interpreted as the binary representation of an integer. Each bit pattern of length 8 corresponds to an automata rule. Therefore it follows immediately that there exist  $2^8 = 256$  different rules (in general:  $k^{k^{2r+1}}$ ). CA in 1D with updating rules depending only on the site itself and the sites immediately adjacent to it on the left and right will be denoted as *elementary cellular automata* (Wolfram, 1983, p. 603). Instead of the tabular form (bit pattern) the automata rules are often referred to by the corresponding integer between 0 and 255 which is called the *rule number*. Thus the rule 00010010 given in Table 2.3.1 is denoted as rule 18. Similar rule number can also be defined for automata with more than two states per cell. Because the number of rules rapidly increases with  $k$  and  $r$  (compare Table 2.3.3) only a small part of all possible rules has been investigated.

**Subclasses of rules.** Subclasses of rules can be obtained by applying the following definitions:

- *Additive* rules:  $f$  is a linear function of its argument modulo  $k$ . Remark: These rules obey a special additive superposition principle and therefore are accessible to an algebraic analysis (Martin et al., 1984).

**Table 2.3.3.** *The number of possible rules for cellular automata with  $k$  states per cell and a range  $r$ . Listed are only the cases where the number is smaller than  $10^{100}$  (Gerling, 1990b).*

$k/r$	1	2	3
2	$2^8$	$2^{32}$	$2^{128}$
3	$3^{27}$	—	—
4	$4^{64}$	—	—
5	$5^{125}$	—	—

- *Totalistic* rules:  $\alpha_j \equiv 1 \quad \forall j$  in (2.3.2), i.e. the cell and all its neighbors in the range  $r$  contribute equally and for  $k = 2$  only the sum of occupied cells matters.
- *Symmetric* rules:  $F[a_{i-r}, \dots, a_{i+r}] = F[a_{i+r}, \dots, a_{i-r}]$ .
- Rules *with memory*:  $a_i^{(t)}$  depends on  $a_i^{(t-1)}$  (otherwise: rules *without memory* or peripheral rules).
- *Legal*<sup>7</sup> rules: rules which do not change the null configuration<sup>8</sup> ("nothing comes of nothing").

**Exercise 2.3.2.** (\*\*)

Cellular automata with  $k = 2$ ,  $r = 1$ :

- How many rules are symmetric?
- How many rules are legal symmetric?
- How many rules are totalistic?
- How many rules have memory?

**Exercise 2.3.3.** (\*\*\*)

Prove the following theorem: All legal symmetric rules of cellular automata with  $k = 2$  and  $r = 1$  form an additive group with elements  $0$ ,  $f_0 = a_{i-1} + a_i + a_{i+1}$ ,  $f_1 = a_{i-1} \cdot a_i + a_i \cdot a_{i+1} + a_{i-1} \cdot a_{i+1}$ ,  $f_2 = a_{i-1} \cdot a_i \cdot a_{i+1}$ ,  $f_3 = (a_i - a_{i-1})(a_i - a_{i+1})$ ,  $f_4 = f_1 \cdot f_3$ .

<sup>7</sup> Please note that some authors require that legal rules should also be symmetric.

<sup>8</sup> The null configuration is the (global) configuration where all cells are empty. In CA with legal rules it is also called the quiescent configuration.

**Cellular automata as a discretization of partial differential equations?** Lattice-gas cellular automata - a special type of cellular automata - are relatively new numerical schemes to solve physical problems ruled by partial differential equations. One could ask whether cellular automata can be interpreted as discrete models of partial differential equations.

Consider the *diffusion equation*

$$\frac{\partial C}{\partial t} = \kappa \frac{\partial^2 C}{\partial x^2} \quad (2.3.3)$$

as an example of a partial differential equations of first order in time. The discretization forward in time and symmetric in space reads

$$C_i^{(t)} = C_i^{(t-1)} + \frac{\Delta t \cdot \kappa}{(\Delta x)^2} \left[ C_{i+1}^{(t-1)} - 2C_i^{(t-1)} + C_{i-1}^{(t-1)} \right] \quad (2.3.4)$$

$$\begin{aligned} &= \sum_{j=-1}^{j=1} \alpha_j C_{i+j}^{(t-1)} \\ &= f \left[ \sum_{j=-1}^{j=1} \alpha_j C_{i+j}^{(t-1)} \right]. \end{aligned} \quad (2.3.5)$$

Here  $f$  is the identity. Eq. (2.3.5) is of the same form as the map (2.3.2) which defines the automata rule. However, there are fundamental differences:

- The coefficients  $\alpha_j$  in (2.3.5) in general are real numbers and not integers.
- The number of states of  $C_i$  is infinite.
- In general  $C_i$  in Eq. (2.3.4) is not bounded whereas the result of  $f()$  in Eq. (2.3.2) is limited to the range 0 to  $k-1$  (modulo constraint).
- Whereas the development in time of the finite number of states is always stable the iteration of (2.3.4) can lead to instability, i.e. the absolute value of the concentration  $C_i$  goes to infinity (try to iterate (2.3.4) with a time step  $\Delta t$  below or slightly above the stability limit  $\Delta t_c = \frac{(\Delta x)^2}{2\kappa}$ ).
- The diffusion equation (and many other partial differential equations in mathematical physics) are based on conservation laws whereas for most of the automata rules no conservation laws are known.

Although there are some formal similarities between discretization of partial differential equations and cellular automata rules the differences dominate. Only special types of cellular automata provide discrete models for partial differential equations of mathematical physics. The connection between the differential equations and lattice gas automata is not formal but deeply rooted in the ground of conservation laws.

**Irreversibility and Garden of Eden configurations.** An important feature of (most) CA is their *local irreversibility*, i.e. under certain local rules different initial (global) configurations may be transformed into the same final configuration. As a consequence of irreversibility not all possible (global) configurations can be reached by time evolution of the CA. The unreachable configurations can only be initialized and therefore are called *Garden of Eden configurations*.

Under most local rules cellular automata behave *locally irreversible*, i.e. different initial configurations are mapped onto the same final configuration. For deterministic rules each configuration has a definite post-configuration (descendant) which can result, however, from several initial configurations (ancestors). Hence the trajectories traced out by the time evolution of several configurations may coalesce, but may never split. A trivial example is provided by a CA with the totalistic null rule: the first iteration transforms arbitrary initial configurations into the null configuration. In a reversible system all configurations have definite post- and pre-configurations. Thus the number of accessible configurations is constant in time (*Liouville's theorem*) and is equal to the number of all possible configurations.

As a consequence of irreversibility there exist configurations that can be initialized but are unreachable during the development in time of the CA. Such configurations are called *Garden of Eden configurations* (Moore, 1962; Aggarwal, 1973). These configurations are not at all seldom. Under the null rule, for example, all configurations except for the null configuration lay in Paradise. Table 2.3.4 gives the fraction of reachable configurations for several rules of elementary cellular automata. Further investigations of Garden of Eden configurations can be found in Voorhees (1990, 1994, 1996), Voorhees and Bradshaw (1994) and Schadschneider and Schreckenberg (1998).

One of the basic decision problems of CA is to decide for a given local rule, whether its global rule has a Garden of Eden (Kutrib et al., 1997). It has been shown to be undecidable for two- and higher-dimensional CA (Kari, 1990; Durand, 1994) while it is decidable for one-dimensional CA (Amoroso and Patt, 1972).

**Exercise 2.3.4. (\*\*)**

How many configurations of the cellular automata with  $N = 10$ ,  $k = 2$ ,  $r = 1$ , periodic boundary conditions, and rule 56 belong to the Garden of Eden?

**Exercise 2.3.5. (\*\*)**

Which rules for cellular automata with  $k = 2$ ,  $r = 1$ , periodic boundary conditions, and  $N = 4$  or  $N = 5$  are reversible?

The irreversible behavior of cellular automata is reflected also in the evolution in time of the information-theoretical (Shannon) entropy  $S$  which is defined



**Table 2.3.4.** *Reachable configurations of elementary cellular automata ( $k = 2$ ,  $r = 1$ ) with periodic boundary conditions; compare Wolfram (1983).  $F_r \leq 1$  is the fraction of reachable configurations (the number of all possible configurations is  $2^N$  where  $N$  is the number of cells).*

Rule	$F_r$	Remarks
0	$1/2^N$	null rule is trivially irreversible
4	$1/2^{N-1}$	no two adjacent sites have the same value
90	$1/2$	if $N$ is odd; even number of cells have value one
90	$1/4$	if $N$ is even
126		depends on $N$ ; $\lim_{N \rightarrow \infty} F_r \rightarrow 0$
204	1	identity transformation is trivially reversible

as usual (but an arbitrary multiplicative constant or a different base for the logarithm<sup>9</sup> can be chosen) by

$$S := - \sum_i p_i \log_2 p_i \quad (2.3.6)$$

(see, for example, Wolfram, 1983)<sup>10</sup>, whereby  $p_i$  is the probability of the (global) configuration  $i$ . The increase in entropy  $S(t)$  with time (compare Exercise 2.3.8) is a reflection of local irreversibility of CA.

**Exercise 2.3.6. (\*)**

Prove:

$$\lim_{x \rightarrow 0} x \log_2 x = 0$$

**Exercise 2.3.7. (\*\*)**

Which distribution  $p_i$  belongs to an extremum of  $S$ ?

**Exercise 2.3.8. (\*\*)**

Calculate  $S(t)$  for  $t = 1$  to 100 for the CA with  $k = r = 2$ , periodic boundary conditions,  $N = 10$  cells, and the totalistic rule 2. The initial ensemble encompasses all possible configurations with equal probabilities.

### 2.3.1 Qualitative characterization of one-dimensional cellular automata

The following rule numbers refer to legal totalistic rules with two states per cell  $k = 2$  and range  $r = 2$ :

<sup>9</sup> The natural logarithm is more appropriate for calculations involving differentiation.

<sup>10</sup> Please note that Wolfram defines the entropy with a different sign:  $S_W := + \sum_i p_i \log_2 p_i$  which actually gives the ‘information content’.

$$a_i^{(t)} = f \left[ \underbrace{\sum_{j=-2}^{j=2} a_{i+j}^{(t-1)}}_{=: s} \right] \quad (2.3.7)$$

The argument  $s$  can take on values between 0 and 5 only<sup>11</sup>. Accordingly a rule is defined by six numbers  $b_i \in \{0, 1\}$ . The sequence  $b_5 b_4 b_3 b_2 b_1 b_0$  can be interpreted as the binary representation of an integer between 0 and 63 which refer to the various totalistic rules. Example: rule  $20 \rightarrow b_5 b_4 b_3 b_2 b_1 b_0 = 010100$

$$a_i^{(t)} = 1 \text{ if } \sum_{j=-2}^{j=2} a_{i+j}^{(t-1)} = 2 \text{ or } 4 \text{ and}$$

$$a_i^{(t)} = 0 \text{ otherwise.}$$

Wolfram (1984a,b) has investigated a large number of one-dimensional automata with legal totalistic rules, two states per cell  $k = 2$ , range  $r = 2$  and random initial conditions. He proposed the following classification<sup>12</sup> with four different types of behavior:

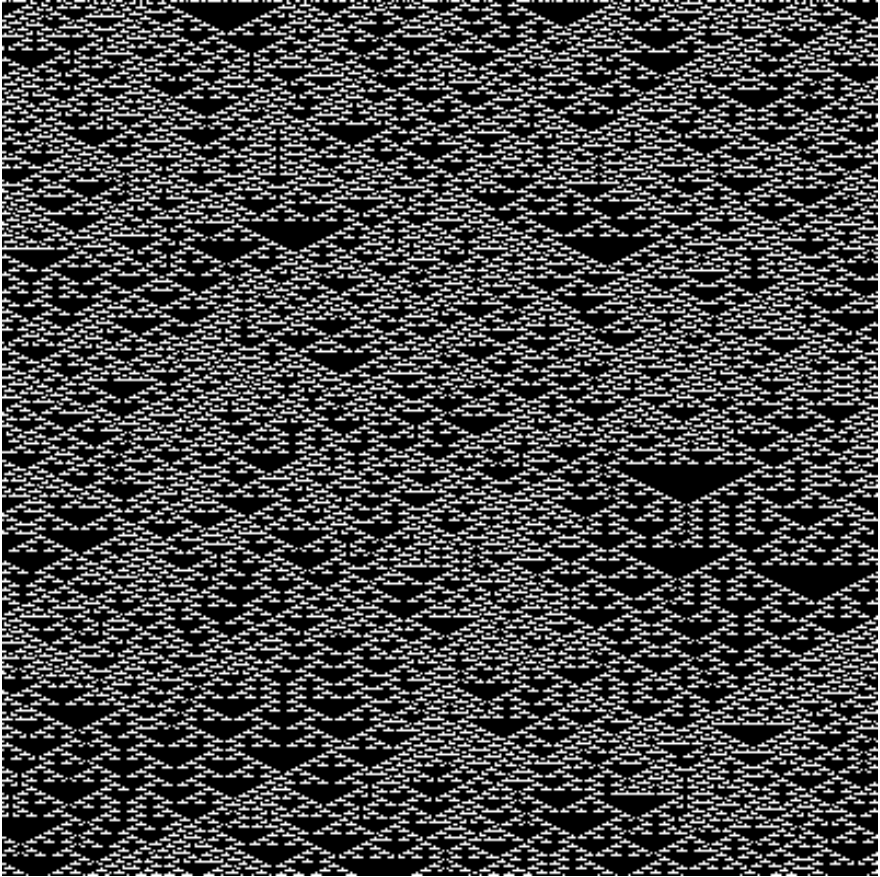
1. The final configuration is homogeneous.  
Rules: 0,4,16,32,36,48,54,60,62.  
Analogue in continuous dynamical systems: limit point.
2. The development in the course of time leads to simple time-independent or time-periodic patterns.  
Rules: 8,24,40,56,58.  
Analogue in continuous dynamical systems: limit cycles.
3. Generation of chaotic patterns.  
Rules: 2,6,10,12,14,18,22,26,28,30,34,38,42,44,46,50.  
Analogue in continuous dynamical systems: strange attractors.
4. The development in the course of time leads to complex local patterns which in part may be long-lived.  
Rules: 20,52.  
There is no analogue in continuous dynamical systems.

The following figures show the development in time of one-dimensional cellular automata with  $k = 2$  possible states per cell, range  $r = 2$ ,  $N = 100$  or  $N = 400$  number of cells, periodic boundary conditions, and legal totalistic rules. The initial configuration (upper line) is set randomly with equal probability to 0 (white) and 1 (black). All figures show the configurations at  $N$  consecutive time levels (from top to bottom).

<sup>11</sup> For arbitrary  $k$  and  $r$ :  $0 \leq s \leq (k-1)(2r+1)$ .

<sup>12</sup> Different classification schemes have been proposed by several authors (Stauffer, 1989; Gerling, 1990a; Binder, 1991; Twining 1992; Cattaneo et al., 1995; Makowiec, 1997).

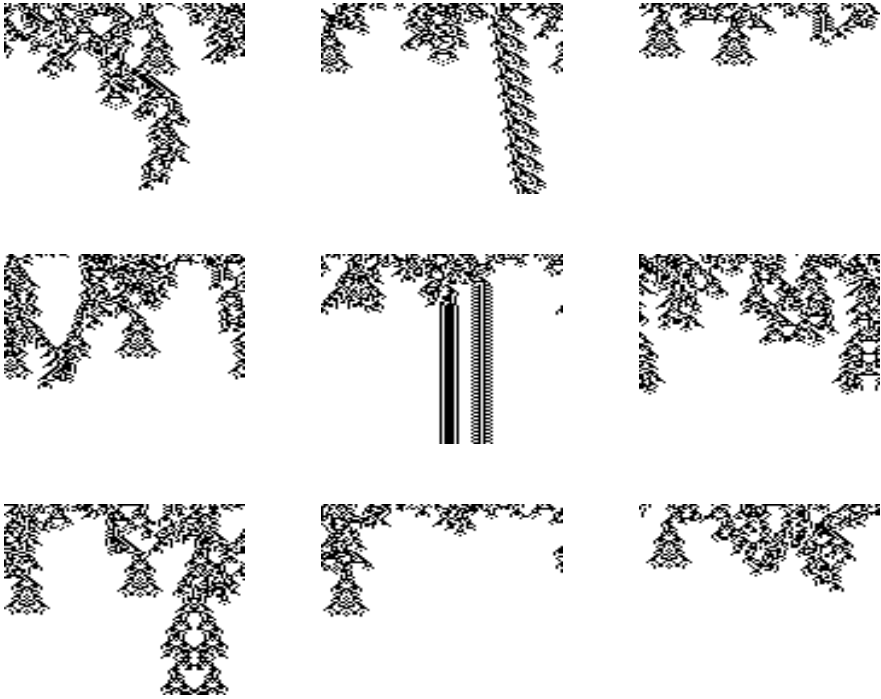
**Fig. 2.3.2.** Cellular automaton with  $k = 2$  possible states per cell, range  $r = 2$ ,  $N = 400$  number of cells, periodic boundary conditions, and random initial configuration (upper line). The figure shows the configurations at 400 consecutive time levels (from top to bottom). The CA with totalistic rule 2 applies under Wolfram's third class. The connection between CA and Sierpinski carpets are discussed, for example, in Wolfram (1983) or Peitgen et al. (1992).



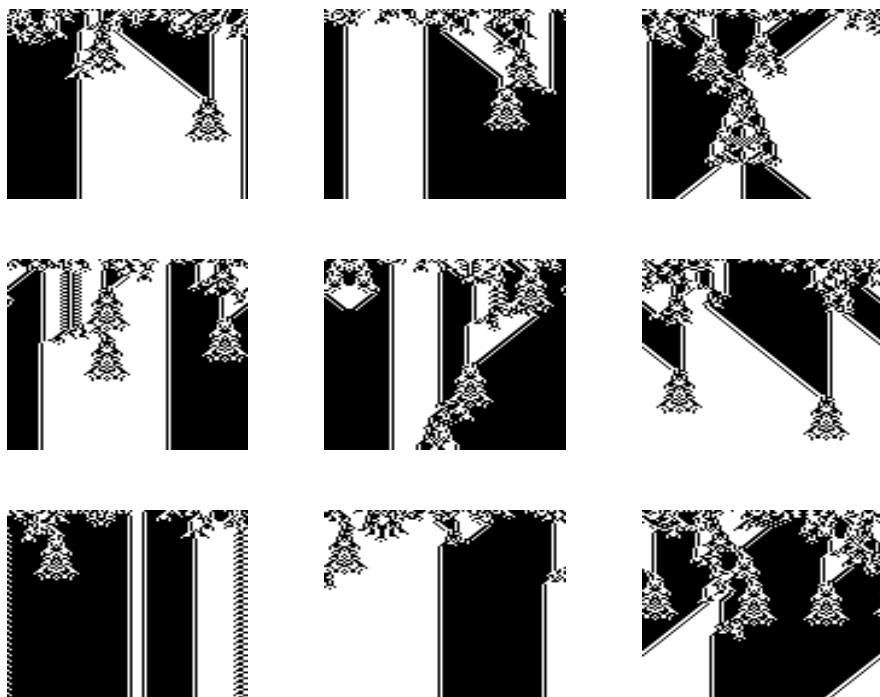
**Table 2.3.5.** *Legal totalistic cellular automata: Classification (approximately!) according to Wolfram (1984b).*

Type	$k = 2$ $r = 1$	$k = 2$ $r = 2$	$k = 2$ $r = 3$	$k = 3$ $r = 1$	
	1	0.5	0.25	0.09	0.12
2	0.25	0.16	0.11	0.19	
3	0.25	0.53	0.73	0.60	
4	0	0.06	0.06	0.07	

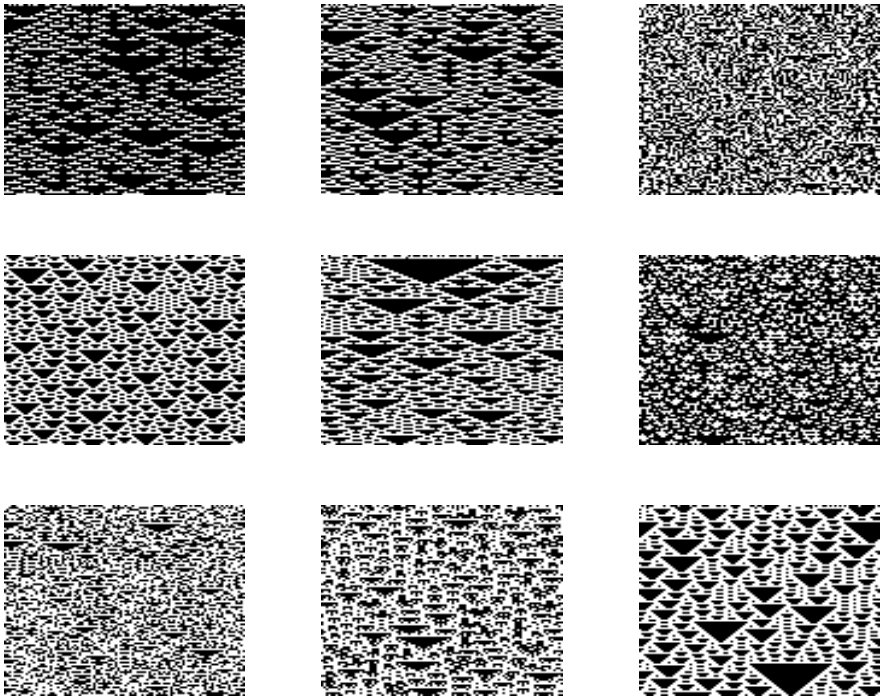
**Fig. 2.3.3.** *1D CA with  $k = 2$ ,  $r = 1$ ,  $N = 100$ , periodic boundary conditions, totalistic rule 20 (Wolfram’s class 4): the nine plots show the configurations at the first hundred time levels starting from different random initial configurations.*



**Fig. 2.3.4.** Same as Fig. (2.3.3) except totalistic rule 52 (Wolfram's class 4).



**Fig. 2.3.5.** *1D CA with  $k = 2$ ,  $r = 1$ ,  $N = 100$ , periodic boundary conditions, totalistic rules 2, 6, 10, 12, 14, 18, 22, 26, 28 (from left to right and from top to bottom; Wolfram's class 3): the nine plots show the configurations at the first hundred time levels starting from the same random initial configuration.*



## 2.4 Two-dimensional cellular automata

In two dimensions there is much more freedom for arranging the cells and defining the neighborhoods for the updating rules. Here only the simplest configurations will be considered. Various other arrangements will be presented in the chapter on lattice-gas cellular automata.

### 2.4.1 Neighborhoods in 2D

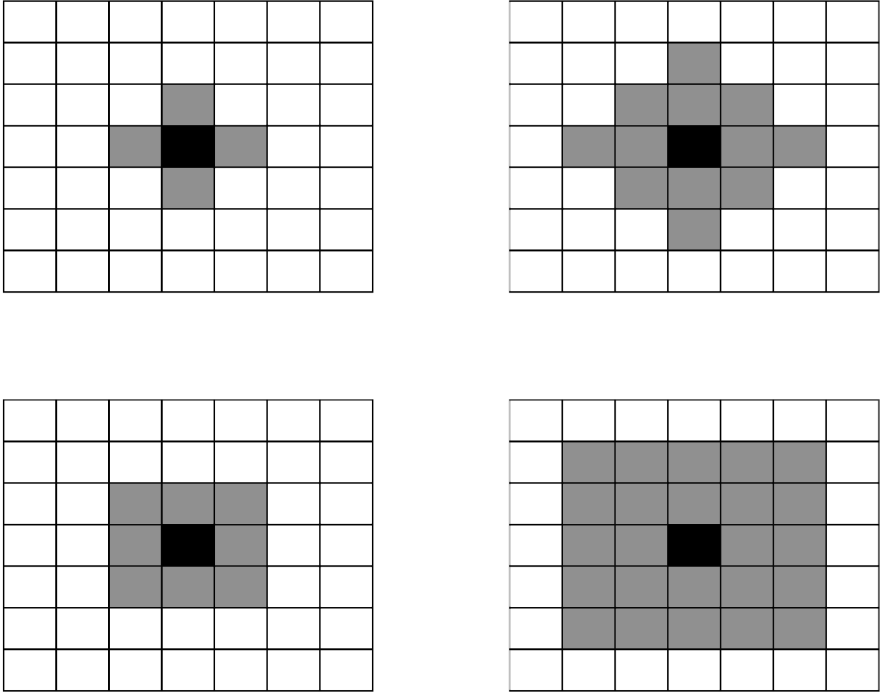
*Von Neumann neighborhoods* of range  $r$  are defined by

$$N_{i,j}^{(vN)} := \{(k, l) \in L \mid |k - i| + |l - j| \leq r\} \quad (2.4.8)$$

and *Moore neighborhoods* of range  $r$  by

$$N_{i,j}^{(M)} := \{(k, l) \in L \mid |k - i| \leq r \text{ and } |l - j| \leq r\}. \quad (2.4.9)$$

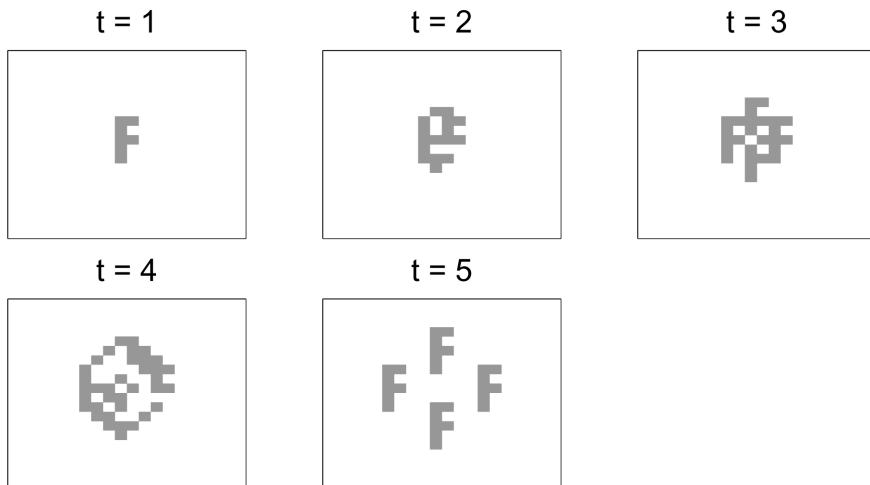
**Fig. 2.4.6.** Neighborhoods in 2D of range 1 and 2: von Neumann (upper), Moore (lower).



### 2.4.2 Fredkin's game

Fredkin proposed a cellular automata game with simple rules which leads to self-replication in a trivial sense, i.e. without configurations that contain *universal Turing machines*. The game is defined as follows (Gardner, 1971b). Each cell has two possible states: alive (occupied) or dead (empty). All cells are updated simultaneously. Count the number of live cells of the four neighbors (von Neumann neighborhood of range 1; compare Fig. 2.4.6). Each cell with an even number (0, 2, 4) of live neighbors will be dead at the next time level and alive otherwise. It can be shown that any initial pattern of live cells will reproduce itself four times after  $2^n$  iterations ( $n$  depends on the initial pattern). The four replicas will be displaced  $2^n$  cells from the vanished original. Fig. 2.4.7 shows an example where  $n = 2$ .

**Fig. 2.4.7.** *Fredkin's game: after  $2^n$  iterations the original pattern of live cells has disappeared and four replicas have shown up at a distance of  $2^n$  cells from the vanished original.  $n$  depends on the initial pattern and is equal to 2 in the example shown here.*





### 2.4.3 ‘Life’

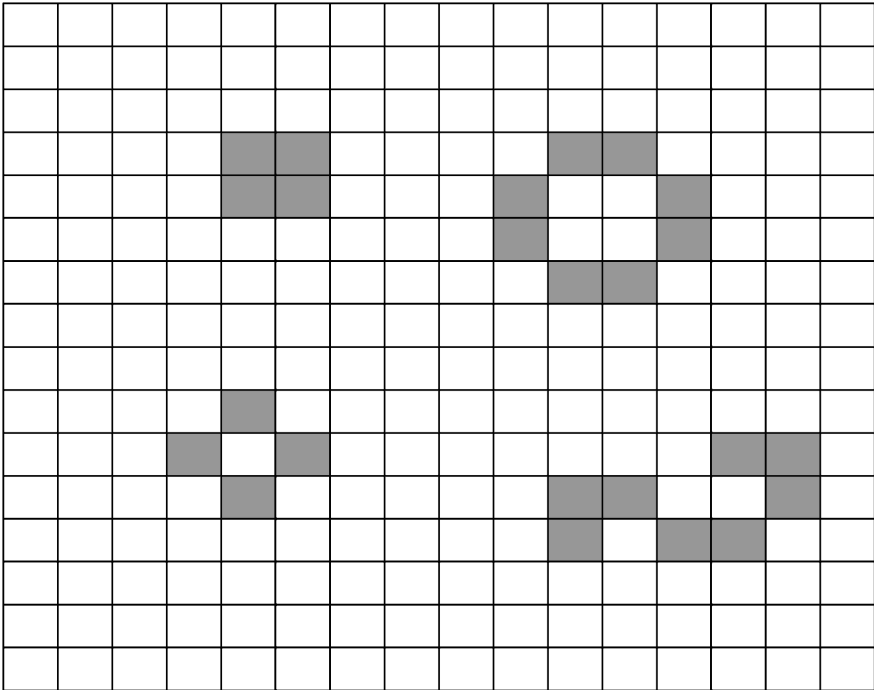
At the beginning of the 70’s Conway introduced the ‘Life’: a two-dimensional synchronous cellular automaton which simulates the evolution of a society of living organisms.

‘Life’ is defined by two rules involving eight neighbors (Moore neighborhood of range 1; compare Fig. 2.4.6):

- Each live site will remain alive the next time-step if it has two or three live neighbors, otherwise it will die.
- At a dead site new live will be born only if there are exactly three live neighbors.

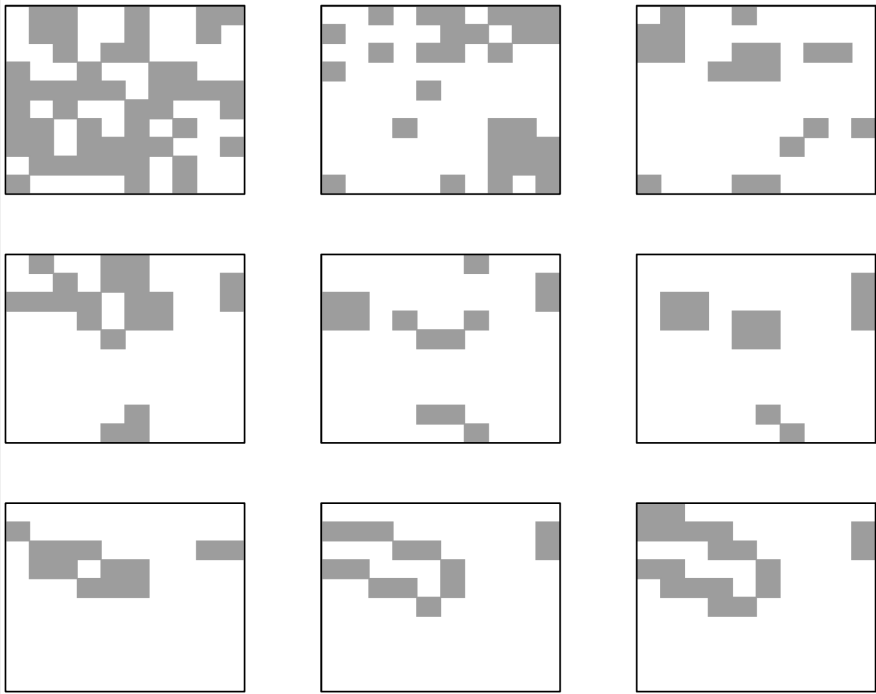
‘Life’ contains many patterns which remain stable from iteration to iteration when not disturbed by other objects (see Fig. 2.4.8 for some examples).

**Fig. 2.4.8.** *The patterns shown here remain stable from generation to generation.*

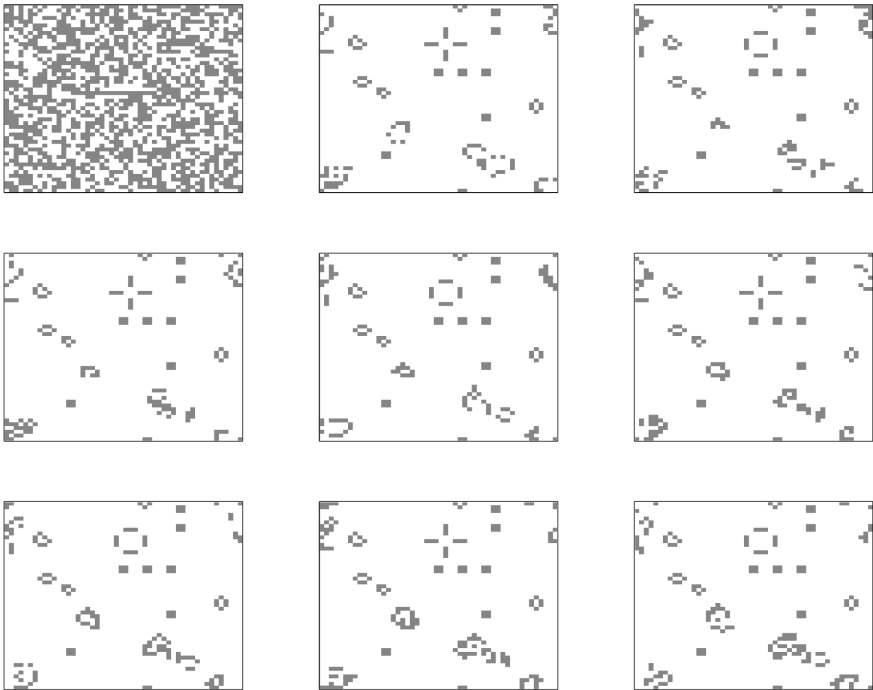


The development in time of initial random configurations with equal probabilities of  $1/2$  for dead or alive is shown in Figures (2.4.9) and (2.4.10). In the limit of large domain size and time approximately 3 % of all cells are alive (compare Fig. 2.4.11).

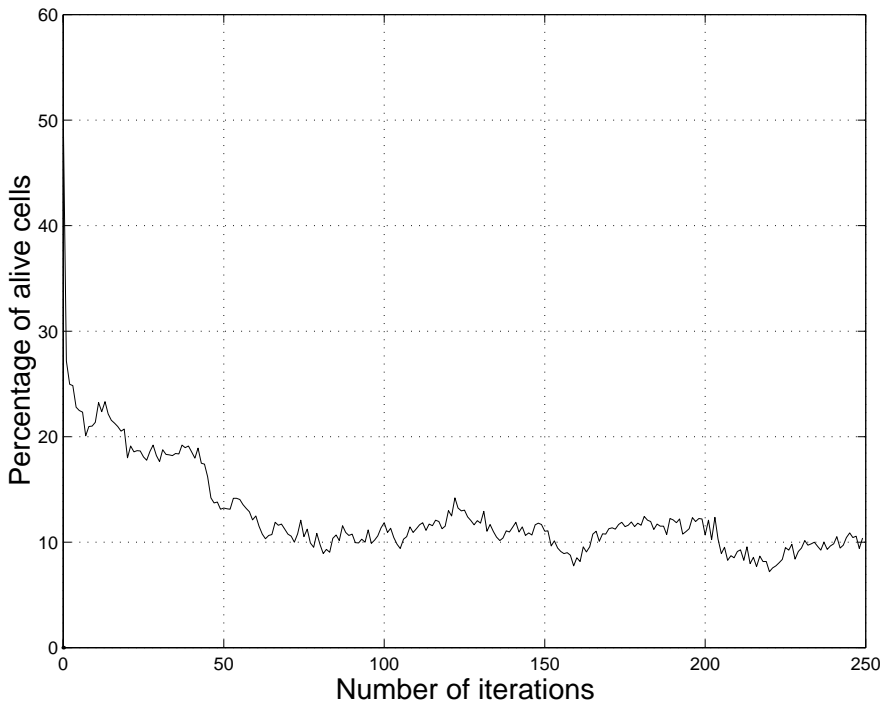
**Fig. 2.4.9.** ‘Life’ on a 10 times 10 array with periodic boundary conditions. The figure shows the random initialization with equal probability for dead or alive (upper left) and the configurations at the eight successive time levels (from left to right and downward).



**Fig. 2.4.10.** ‘Life’ on a 50 times 50 array with periodic boundary conditions. Upper left: Random initialization with equal probability for dead or alive. The other plots show the configurations at time levels 141 to 148.



**Fig. 2.4.11.** ‘Life’ on a 50 times 50 array with periodic boundary conditions: percentage of alive cells as a function of time (iterations). The limit for large domain size and time is not yet reached.



**Exercise 2.4.1. (\*\*)**

Find all stable configurations with an extension of 3 times 3 cells at most.

**Exercise 2.4.2. (\*)**

The rule of Conway's '*Life*' is one of many possible rules for a two-state cellular automaton in two-dimensions when the rule takes into account the states of the nearest neighbors and the cell itself. One can wonder whether the self-organized critical regime can occur for '*Life*' only or can occur for many other of the possible rules. Suppose a supercomputer needs only one microsecond to investigate one single rule. How long does it take to analyse all the possible rules?

**Further reading:**

Sigmund (1993), Alstrøm and Leão (1994), Vandewalle and Ausloos (1995), Bak (1996), Heudin (1996), Pulsifer and Reiter (1996), Nordfalk and Alstrøm (1996), de la Torre and Martin (1997), Malarz et al. (1998).

**2.4.4 CA: what else? Further reading**

Books, proceedings, reviews:

- Codd (1968)
- Burks (1970)
- Wolfram (1986)
- Jackson (1990, chapter 10)
- Gutowitz (1991),
- Wolfram (1994)
- Vollmar et al. (1996)
- Voorhees (1996)
- Kutrib et al. (1997)
- 2nd Conference on Cellular Automata for Research and Industry, Theoretical Computer Science, 217(1), 1999.

Articles:

- Invertible CA: Toffoli and Margolus (1990).
- Elementary reversible CA: Takesue (1987, 1989, 1990).
- Additive invariants: Hattori and Takesue (1991).
- Staggered invariants: Takesue (1995).

- Normal forms of CA: see references in Kutrib et al. (1997).
- Fractals: Peitgen et al. (1998).
- Simulation of traffic flow with cellular automata: Nagel and Schreckenberg (1992), Schreckenberg et al. (1995), Benjamin et al. (1996), Brankov et al. (1996), Chopard et al. (1996), Fukui and Ishibashi (1996a,b), Ishibashi and Fukui (1996a,b), Kerner et al. (1996), Rickert et al. (1996), Barlovic et al. (1998), Emmerich et al. (1998), Huang (1998), Nagel et al. (1998), Schadschneider and Schreckenberg (1998), Wang et al. (1998).
- Biology: Stuart A. Kauffman (1984, 1986, 1991) applied CA with complex interconnection to biological problems; Ermentrout and Edlestein-Keshet (1993) give a review of biologically motivated CA. Bandini et al. (1998), Bastolla and Parisi (1998a,b), Mielke and Pandey (1998), Siregar et al. (1996, 1998), Zorzenon Dos Santos (1998), O'Toole et al. (1999).
- Statistical mechanics: Ruján (1987)
- Pattern formation: Lindgren et al. (1998).
- Quantum cellular automata: Watrous (1995), Richter and Werner (1996), Tougaw and Lent (1996)
- More papers can be found in the following journals:
  - Complex Systems
  - Physica D (for example: volume 45, p. 3-479, 1990 and volume 103, 1997).

#### 2.4.5 From CA to LGCA

Despite of their simple update rules cellular automata can display complex behavior which is a prerequisite to use them as a simulation tool for physical (biological, chemical, ...) phenomena like, for example, fluid flow. CA are very easy to implement and are especially well suited for massively parallel computers because of the local character of the update rules. By construction they are unconditionally numerically stable.

Before CA are to be used for the simulation of physical processes the following items have to be addressed:

1. Many physical laws are based on the conservation of certain quantities. The Navier-Stokes equation, for example, expresses the conservation of mass and momentum. The cellular automata used for simulation should hold corresponding conserved quantities. As will be shown later on one of the main problems in the construction of lattice-gas cellular automata is to avoid the occurrence of additional (non-physical or spurious) invariants.

2. Nonstatic physical phenomena involve the transport of certain quantities. The propagation of information in CA is not possible for most rules. Among the elements of the group  $G = (0, f_0, f_1, f_2, f_3, f_4; +)$  introduced in Exercise 2.3.3, for example, only one special element ( $f_0$ ) can generate propagation.
3. The desired physical behavior of a lattice-gas cellular automata will show up in the macroscopic limit which can be derived from a theory of statistical mechanics on a lattice. The application of certain concepts of statistical mechanics requires that the microdynamics, i.e. the update rules, are *reversible*.

Only a small subset of CA holds the appropriate number of conserved quantities, is able to propagate these quantities or has reversible rules. Based on the discussion of CA given above it is not clear whether or not CA exist with all three properties and if so, how to construct such CA for a given phenomenon.

In order to simplify the problem of constructing cellular automata for given physical processes lattice-gas cellular automata (LGCA) differ somewhat from the CA discussed above in that the update is split into two parts which are called *collision* and *propagation* (or *streaming*). The collision rule of LGCA can be compared with the update rule for CA in that it assigns new values to each cell based on the values of the cells in a local neighborhood. After the collision step the state of each cell is propagated to a neighboring cell. This split of the update guarantees propagation of quantities while keeping the proper update rules (collisions) simple. Because of this difference of LGCA from the CA discussed here so far some authors do not consider LGCA as proper CA and prefer to speak of *lattice gases* (see, for example, Hénon, 1989b). This latter notation might, however, lead to confusion. “There are at least two, almost independent, lattice gas communities: one community ... usually moves bits or numbers around a lattice while conserving momentum; the other community, mostly solid-state theorists, focuses almost exclusively on the Ising model.” (Quoted from J. Stat. Phys., Vol. 68, Nos. 3/4, p. 611, 1992.)