Cellular Automata Modeling of Physical Systems

Book · January 1998

DOI: 10.1007/978-0-387-30440-3_57

2 authors:

Bastien Chopard
University of Geneva
272 PUBLICATIONS **6,041** CITATIONS

SEE PROFILE

Michel Droz
University of Geneva
180 PUBLICATIONS **4,994** CITATIONS

SEE PROFILE

Cellular Automata Modeling of Physical Systems

Bastien Chopard and Michel Droz
University of Geneva

CAMBRIDGE
UNIVERSITY PRESS

PUBLISHED BY THE PRESS SYNDICATE OF THE UNIVERSITY OF CAMBRIDGE
The Pitt Building, Trumpington Street, Cambridge CB2 1RP, United Kingdom

CAMBRIDGE UNIVERSITY PRESS
The Edinburgh Building, Cambridge CB2 2RU, UK http://www.cup.cam.ac.uk
40 West 20th Street, New York, NY 10011-4211, USA http://www.cup.org
10 Stamford Road, Oakleigh, Melbourne 3166, Australia

First published 1998

Printed in the United Kingdom at the University Press, Cambridge

Typeset in 11pt Times [EPC]

A catalogue record for this book is available from the British Library

Library of Congress Cataloguing in Publication data

Chopard, Bastien, 1959–
Cellular automata modeling of physical systems / Bastien Chopard and Michel Droz.
p. cm. – (Collection Aléa-Saclay)
Includes bibliographical references and index.
ISBN 0 521 46168 5 (hc: alk. paper)
1. Statistical physics. 2. Cellular automata. 3. Diffusion.
4. Lattice gas. 5. Phase transformations (Statistical physics)
I. Droz, M. (Michel), 1944– . II. Title. III. Series.
QC174.8.C49 1998
530'.13'01'1–dc21 97-28284 CIP

ISBN 0 521 46168 5 hardback

Contents

1
Introduction

1.1 Brief history

Cellular automata (often termed CA) are an idealization of a physical system in which space and time are discrete, and the physical quantities take only a finite set of values.

Although cellular automata have been reinvented several times (often under different names), the concept of a cellular automaton dates back from the late 1940s. During the following fifty years of existence, cellular automata have been developed and used in many different fields. A vast body of literature is related to these topics. Many conference proceedings [1–8]), special journal issues [9,10] and articles are available.

In this section, our purpose is not to present a detailed history of the developments of the cellular automata approach but, rather, to emphasize some of the important steps.

1.1.1 Self-reproducing systems

The reasons that have led to the elaboration of cellular automata are very ambitious and still very present. The pioneer is certainly John von Neumann who, at the end of the 1940s, was involved in the design of the first digital computers. Although von Neumann's name is definitely associated with the architecture of today's sequential computers, his concept of cellular automata constitutes also the first applicable model of massively parallel computation.

Von Neumann was thinking of imitating the behavior of a human brain in order to build a machine able to solve very complex problems. However, his motivation was more ambitious than just a performance increase of the computers of that time. He thought that a machine with such a complexity as the brain should also contain self-control and self-repair mechanisms. His idea was to get rid of the difference which exists between processors and the data, by considering them on the same footing. This led him to envisage a machine capable of building itself, out of some available material.

Rapidly, he considered the problem from a more formal viewpoint and tried to define the properties a system should have to be self-replicating. He was mostly interested to find a logical abstraction of the self-reproduction mechanism, without reference to the biological processes involved.

Following the suggestions of S. Ulam [11], von Neumann addressed this question in the framework of a fully discrete universe made up of cells. Each cell is characterized by an internal state, which typically consists of a finite number of information bits. Von Neumann suggested that this system of cells evolves, in discrete time steps, like simple automata which only know of a simple recipe to compute their new internal state. The rule determining the evolution of this system is the same for all cells and is a function of the states of the neighbor cells. Similarly to what happens in any biological system, the activity of the cells takes place simultaneously. However, the same clock drives the evolution of each cell and the updating of the internal state of each cell occurs synchronously. These fully discrete dynamical systems (cellular space) invented by von Neumann are now referred to as *cellular automata.*

The first self-replicating cellular automaton proposed by von Neumann was composed of a two-dimensional square lattice and the self-reproducing structure was made up of several thousand elementary cells. Each of these cells had up to 29 possible states [12]. The evolution rule required the state of each cell plus its four nearest neighbors, located north, south, west and east. Due to its complexity, the von Neumann rule has only been partially implemented on a computer [13].

However, von Neumann had succeeded in finding a discrete structure of cells bearing in themselves the recipe to generate new identical individuals. Although this result is hardly even a very primitive form of life, it is quite interesting because it is usually expected that a machine can only build an object of lesser complexity than itself. With self-replicating cellular automata, one obtains a "machine" able to create new machines of identical complexity and capabilities.

The von Neumann rule has the so-called property of universal computation. This means that there exists an initial configuration of the cellular automaton which leads to the solution of any computer algorithm. This sounds a surprising statement: how will such a discrete dynamics help us to solve any problem? It turns out that this property is of theoretical rather than practical interest. Indeed, the property of universal computing means that any computer circuit (logical gates) can be simulated by the rule of the automaton. All this shows that quite complex and unexpected behavior can emerge from a cellular automaton rule.

After the work of von Neumann, others have followed the same line of research and the problem is still of interest [14]. In particular, E.F. Codd [15] in 1968 and much later C.G. Langton [16] and Byl [17] proposed much simpler cellular automata rules capable of self-replicating and using only eight states. This simplification was made possible by giving up the property of computational universality, while still conserving the idea of having a spatially distributed sequence of instructions (a kind of cellular DNA) which is executed to create a new structure and then entirely copied in this new structure.

More generally, artificial life is currently a domain which is intensively studied. Its purpose is to better understand real life and the behavior of living species through computer models. Cellular automata have been an early attempt in this direction and can certainly be further exploited to progress in this field [18,19].

1.1.2 Simple dynamical systems

In a related framework, it is interesting to remember that it is precisely a simple ecological model that has brought the concept of cellular automata to the attention of wide audience. In 1970, the mathematician John Conway proposed his now famous *game of life* [20]. His motivation was to find a simple rule leading to complex behaviors. He imagined a two-dimensional square lattice, like a checkerboard, in which each cell can be either alive (state one) or dead (state zero). The updating rule of the game of life is as follows: a dead cell surrounded by exactly three living cells comes back to life. On the other hand, a living cell surrounded by less than two or more than three neighbors dies of isolation or overcrowdness. Here, the surrounding cells correspond to the neighborhood composed of the four nearest cells (north, south, east and west) plus the four second nearest neighbors, along the diagonals. Figure 1.1 shows three configurations of the game of life automaton, separated by 10 iterations.

It turned out that the game of life automaton has an unexpectedly rich behavior. Complex structures emerge out of a primitive "soup" and evolve so as to develop some skills. For instance, objects called *gliders* may form (see problems, section 1.4). Gliders correspond to a particular arrangement of adjacent cells that has the property to move across space, along straight trajectories. Many more such structures have been identified in the vast body of literature devoted to the game of life [21,22]. As for the von Neumann rule, the game of life is a cellular automata capable of computational universality.

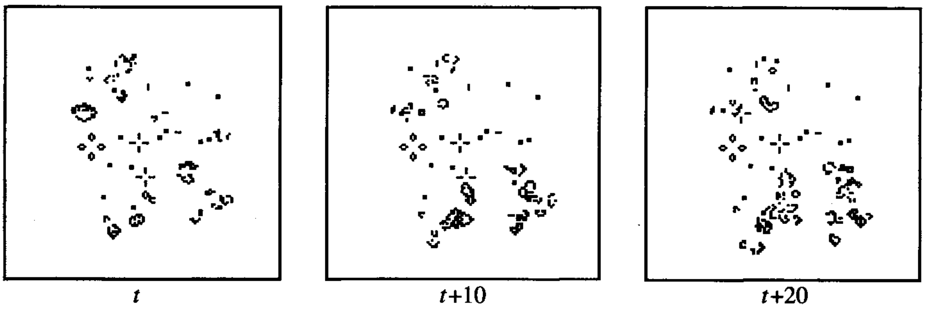

Fig. 1.1. The game of life automaton. Black dots represents living cells whereas dead cells are white. The figure shows the evolution of some random initial configurations.

In addition to these theoretical aspects, cellular automata were used in the 1950s for image processing [23]. It was recognized early on that much tedious picture analysis could be carried out automatically, according to a cellular automata computing model: the pixels of an image can be treated simultaneously, using simple local operations. Special-purpose machines based on cellular automata logic have been developed for noise reduction, counting and size estimation in images obtained from observations with a microscope.

At the beginning of the 1980s, S. Wolfram studied in detail a family of simple one-dimensional cellular automata rules (the now famous Wolfram rules [24,25]). He had noticed that a cellular automaton is a discrete dynamical system and, as such, exhibits many of the behaviors encountered in a continuous system, yet in a much simpler framework. A concept such as complexity could be investigated on mathematical models allowing an exact numerical computer calculation, because of their Boolean nature (no numerical errors nor truncation as in more traditional models). Wolfram's results have contributed to prove that cellular automata are important objects to consider for statistical mechanics studies and, at the present time, Wolfram's rule are still the topic of much research.

1.1.3 A synthetic universe

The property of many cellular automata rules being a universal computer made several authors think that the physical world itself could be a very large cellular automaton. Tommaso Toffoli [26] compares cellular automata to a synthetic model of the universe in which the physical laws are expressed in terms of simple local rules on a discrete space–time structure.

T. Toffoli, N. H. Margolus and E. Fredkin recognized the importance of cellular automata as a modeling environment for physical systems. They were very interested in the analogy that exists between the theory of information as it is used to describe numerical processing in a computer and the laws of physics. Cellular automata provide an excellent framework to develop these ideas. In particular, they showed how to build a fully time-reversible logic from which any numerical operation can be implemented without any loss of information. The so-called *billiard ball* [26] is a cellular automata rule which is an example of such a reversible model of computation.

The possibility of displaying, on a computer screen, the time evolution of large cellular automata systems, at the rate of several updates per second of the complete lattice offers a way of performing experiments live on an artificial universe, whose evolution rules are set up by the observer. By building their first general purpose cellular automata machines CAM-6 in the mid-1980s, Toffoli and Margolus provided a very powerful cellular automata environment with the capability of a supercomputer of that time, at a very affordable price and with unique display facilities. This machine has stimulated many developments of cellular automata techniques and has contributed to the spreading of the main ideas to a wide audience of scientists.

Toffoli and Margolus's book [26]: *Cellular Automata Machines: a New Environment for Modeling,* is a wonderful source of inspiration in the field of cellular automata and provide a complete description of the CAM-6 hardware. More recently, Toffoli, Margolus and coworkers have designed CAM-8, a much more powerful hardware environment: a parallel, uniform, scalable architecture for cellular automata experimentation [27]. This hardware platform offers high performance, a flexible approach, display facilities and is naturally appropriate to work on three-dimensional systems. It has been successfully used for many different applications.

1.1.4 Modeling physical systems

It was also in the 1980s that an important step in the theory of cellular automata was accomplished. It was recognized that the so-called HPP [28] lattice gas models developed in the 1970s by Hardy, Pomeau and de Pazzis was in fact a cellular automata. This model consists of a simple and fully discrete dynamics of particles moving and colliding on a two-dimensional square lattice, in a such a way as to conserve momentum and particle number.

The HPP dynamics was initially planned as a theoretical model to study fundamental statistical properties of a gas of interacting particles. The actual implementation of this model as a cellular automata rule and the visualization of the fast moving particle shed a different light on the possibilities of such models: isn't it possible to simulate the behavior of a real system of particles (like a fluid or a gas) as a cellular automata rule? After all, it is well known that the flows of a fluid, a gas or even a granular medium are very similar at a macroscopic scale, in spite of their different microscopic nature. A fully discrete and simplified molecular dynamics could work too, provided the system is considered at an appropriate observation scale.

Of course, the idea of using discrete systems as a model of real phenomena has already been considered for several problems. The Ising model of classical spin is a famous example which will be discussed in more detail in the next chapter. From the fluid side, already at the end of the nineteenth century, Maxwell [29], had proposed a discrete velocity system of interacting particles as a model of a gas. In fact, such *lattice gas*, discrete velocity models have been developed independently from cellular automata theory [30,31].

However, cellular automata provide a new conceptual framework, as well as an effective numerical tool, which retains important aspects of the microscopic laws of physics, such as simultaneity of the motion, locality of the interactions and time reversibility.

Cellular automata rules are viewed as an alternative form of the microscopic reality which bears the expected macroscopic behavior. From a numerical point of view it was expected, at the end of the 1980s, that a wind tunnel could be replaced by a fully discrete computer model. The first cellular automata model to give credit to this possibility is the famous FHP model proposed in 1986 by U. Frisch, B. Hasslacher and Y. Pomeau [32], and almost simultaneously by S. Wolfram [33]. These authors showed that their model, despite its fully discrete dynamics, follows, in some appropriate limits, the behavior prescribed by the Navier–Stokes equation of hydrodynamics.

Note that models like FHP or HPP are often termed lattice gas automata (LGA) to distinguish them from the less specific cellular automata terminology. Clearly, from a mathematical point of view, a lattice gas automata is a cellular automata, but the way one thinks, for instance, of the game of life is quite different from the underlying philosophy of the FHP model. This difference will become clear to the reader as he or she becomes more familiar with the next chapter of this book. Nevertheless, in this book, we will often use cellular automata to designate a LGA.

Since the FHP rule was discovered, lattice gas automata or cellular automata fluids as these kind of particle models are now often referred to, have been developed intensively and several insufficiencies of the initial model corrected. The Ecole Normale Supérieure in Paris has been very active and P. Lallemand and D. d'Humieres, in particular, have played a pioneering role in this field [34–37].

However, contrary to first expectations, lattice gas models of fluids have not been able to surpass the traditional numerical methods of hydrodynamics and compute high Reynolds flows. Their relatively high viscosity, which is only determined by the cellular automata rule (and therefore not adjustable), is a limiting factor to the practical study of many of these flows. The finite spatial resolution of the cellular automata lattice (physical phenomena must occur at a much larger scale than the lattice spacing) is another limitation on the study and modeling of fully developed turbulence, unless the system has such a large scale that the advantage of a cellular automata approach vanishes even on today's fastest computers [38].

However, lattice gas automata have been much more successful in modeling complex situations for which traditional computing techniques are not applicable. Flows in porous media [39–41], immiscible [42–46] flows and instabilities, spreading of a liquid droplet and wetting phenomena [47], microemulsion [48] erosion and transport problems [49] are some examples pertaining to fluid dynamics.

Other physical situations, like pattern formation, reaction-diffusion processes [50], nucleation–aggregation growth phenomena, are very well described by cellular automata dynamics and will be investigated in detail in this book.

1.1.5 *Beyond the cellular automata dynamics: lattice Boltzmann methods and multiparticle models*

Very often, the advantage of the cellular automata (or lattice gas) approach is most apparent when complex boundary conditions are present. Due to the microscopic interpretation of the dynamics, these conditions can be taken into account in a much more natural way than in a continuous description (like a differential equation) in which our basic intuition of the phenomena may be lost.

On the other hand, cellular automata models have several weakness related to their fully discrete nature: statistical noise requiring systematic averaging processes, and little flexibility to adjust parameters of a rule in order to describe a wider range of physical situations. At the end of the 1980s, McNamara and Zanetti [51], and Higueras, Jimenez and Succi [52] showed the advantage of extending the Boolean dynamics of the automaton to directly work on real numbers representing the probability for a cell to have a given state.

This approach, called the *lattice Boltzmann method* (LBM), is numerically much more efficient than the Boolean dynamics and provides a new computational model much more suited to the simulation of high Reynolds flows and many other relevant applications (for instance glacier flow [53]).

Lattice Boltzmann models retain the microscopic level of interpretation of the cellular automata approach but neglect many-body correlation functions. However, this method now constitutes a very promising approach to modeling physical systems and is discussed on several occasions throughout this book.

In between the strict cellular automata approach and the more flexible lattice Boltzmann method, there is room for an intermediate description: the multiparticle models which are still under development at the present time. These models preserve the concept of a quantized state but an infinite set of values is accepted. Consequently, numerical stability is guaranteed (as opposed to the LBM), and many-body correlations taken into account. The large number of possible states offers more flexibility when modeling a physical system and yields less statistical noise. But a multiparticle dynamics is more difficult to devise and numerically slower than its lattice Boltzmann counterpart. Examples of this approach will be presented in this book.

From our point of view, the cellular automata approach is not a rigid framework. It is rather a philosophy of modeling which should be considered with some pragmatism. The important issue in cellular automata modeling is to capture the essential features of given phenomena and translate them to a suitable form to obtain an effective numerical model. To this end, it is acceptable (and even beneficial) to relax some of the constraints of the original definition of a cellular automata. The introduction of the lattice Boltzmann method is an illustration of this fact. The point is to conserve the spirit of the approach and its relevant features rather than its limitations. This remark is particularly in order because present parallel computers offer an ideal and quite flexible platform to implement cellular automata models without the restrictions imposed by dedicated hardware.

1.2 A simple cellular automaton: the parity rule

In this section, we discuss a simple cellular automata rule, in order to introduce and illustrate the concept. This should slowly familiarize the reader with a more precise notion of cellular automata. Section 1.3 will present a more formal definition.

Although it is very basic, the rule we study here exhibits a surprisingly rich behavior. It was proposed initially by Edward Fredkin in the 1970s [54] and is defined on a two-dimensional square lattice.

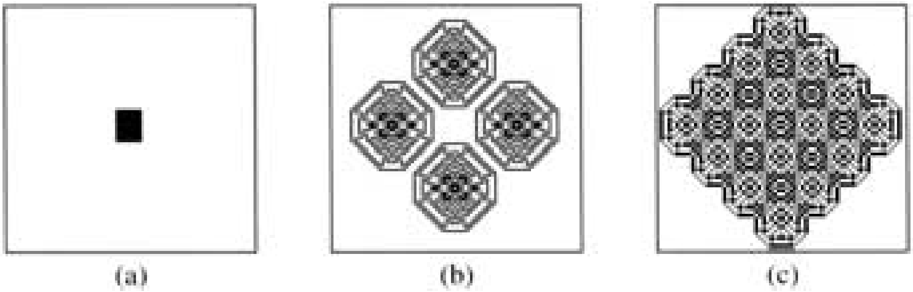

Fig. 1.2. The $\oplus$ rule on a 256×256 periodic lattice: (a) initial configuration; (b) and (c) configurations after $t_b = 93$ and $t_c = 110$ iterations, respectively.

Each site of the lattice is a cell which is labeled by its position $\vec{r} = (i, j)$ where i and j are the row and column indices. A function $\psi_t(\vec{r})$ is associated to the lattice to describe the state of each cell at iteration t. This quantity can be either 0 or 1.

The cellular automata rule specifies how the states ψ_{t+1} are to be computed from the states at iteration t. We start from an initial condition at time $t = 0$ with a given configuration of the values $\psi_0(\vec{r})$ on the lattice. The state at time $t = 1$ will be obtained as follows

(1) Each site $\vec{r}$ computes the sum of the values $\psi_0(\vec{r}\ ')$ on the four nearest neighbor sites $\vec{r}\ '$ at north, west, south and east. The system is supposed to be periodic in both i and j directions (as on a torus) so that this calculation is well defined for all sites.

(2) If this sum is even, the new state $\psi_1(\vec{r})$ is 0 (white), else, it is 1 (black).

The same rule (step 1 and 2) is repeated to find the states at time $t = 2, 3, 4, \ldots$.

From a mathematical point of view, this cellular automata parity rule can be expressed by the following relation

$$\psi_{t+1}(i, j) = \psi_t(i+1, j) \oplus \psi_t(i-1, j) \oplus \psi_t(i, j+1) \oplus \psi_t(i, j-1) \qquad (1.1)$$

where the symbol $\oplus$ stands for the exclusive OR logical operation. It is also the sum modulo 2: $1 \oplus 1 = 0 \oplus 0 = 0$ and $1 \oplus 0 = 0 \oplus 1 = 1$.

When this rule is iterated, very nice geometrical patterns are observed, as shown in figure 1.2. This property of generating complex patterns starting from a simple rule is actually generic of many cellular automata rules. Here, complexity results from some spatial organization which builds up as the rule is iterated. The various contributions of successive iterations combine together in a specific way. The spatial patterns that are observed reflect how the terms are combined algebraically.

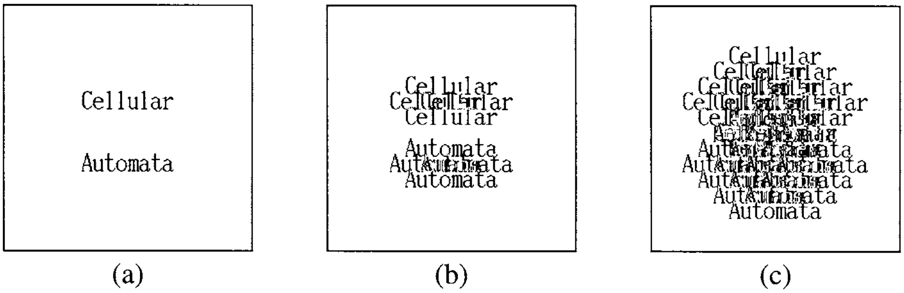

Fig. 1.3. The $\oplus$ rule replicates any initial pattern when the number of iterations is a power of two. Image (a) shows the initial pattern at time $t_a = 0$. Images (b) and (c) show successive iterations at times $t_b = 16$ and $t_c = t_b + 32$.

On closer inspection, we also observe that the initial pattern is replicated at some specific iteration. Figure 1.3 illustrates that point with a more enlightening initial condition. The times at which this happens are a power of two. Another surprising fact occurs when the system size L is a power of two: after $L/2$ iteration the state of each cell vanishes for all possible initial configurations.

These behaviors, as well as the way the pattern builds up can be explained by working out the definition of the rule. Applying the rule 1.1 twice yields ψ_{t+1} as a function of ψ_{t-1}

$$\begin{aligned}\psi_{t+1}(i,j) &= \psi_{t-1}(i+2,j) \oplus \psi_{t-1}(i,j) \oplus \psi_{t-1}(i+1,j+1) \\ &\oplus \psi_{t-1}(i+1,j-1) \oplus \psi_{t-1}(i,j) \oplus \psi_{t-1}(i-2,j) \\ &\oplus \psi_{t-1}(i-1,j+1) \oplus \psi_{t-1}(i-1,j-1) \oplus \psi_{t-1}(i+1,j+1) \\ &\oplus \psi_{t-1}(i-1,j+1) \oplus \psi_{t-1}(i,j+2) \oplus \psi_{t-1}(i,j) \\ &\oplus \psi_{t-1}(i+1,j-1) \oplus \psi_{t-1}(i-1,j-1) \oplus \psi_{t-1}(i,j) \\ &\oplus \psi_{t-1}(i,j-2) \end{aligned} \qquad (1.2)$$

Since $a \oplus a = 0$ and $a \oplus 0 = a$, for all values of a, one obtains

$$\psi_{t+2}(i,j) = \psi_t(i+2,j) \oplus \psi_t(i-2,j) \oplus \psi_t(i,j+2) \oplus \psi_t(i,j-2) \qquad (1.3)$$

Thus, after two iterations, the action of the rule is to translate the initial configuration by two lattice sites in the four directions and XOR them.

If now we compute similarly $\psi_{t+3}(i,j)$ as a function of $\psi_t(i,j)$, we no longer get such a simple expression. A relation similar to equation 1.2 is obtained, but without any cancellation. We see that $\psi_{t+3}(i,j)$ is a superposition of 16 translations. As a result, we obtain the rich geometrical structure observed in figure 1.2.

However, it is easy to prove that the behavior of the $\oplus$ rule is simple when the number of iterations performed is a power of two. To show this property, suppose the following relation is true for a given value of T.

$$\psi_t(i,j) = \psi_{t-T}(i+T,j)\oplus\psi_{t-T}(i-T,j)\oplus\psi_{t-T}(i,j+T)\oplus\psi_{t-T}(i,j-T) \quad (1.4)$$

We already know it is true for $T = 1$ and $T = 2$. Then, if we apply this same relation to ψ_{t-T} on the right-hand side, we obtain

$$\begin{aligned}\psi_t(i,j) &= \psi_{t-2T}(i+2T,j)\oplus\psi_{t-2T}(i,j)\oplus\psi_{t-2T}(i+T,j+T)\\ &\quad\oplus\psi_{t-2T}(i+T,j-T)\oplus\psi_{t-2T}(i,j)\oplus\psi_{t-2T}(i-2T,j)\\ &\quad\oplus\psi_{t-2T}(i-T,j+T)\oplus\psi_{t-2T}(i-T,j-T)\\ &\quad\oplus\psi_{t-2T}(i+T,j+T)\oplus\psi_{t-2T}(i-T,j+T)\oplus\psi_{t-2T}(i,j+2T)\\ &\quad\oplus\psi_{t-2T}(i,j)\oplus\psi_{t-2T}(i+T,j-T)\oplus\psi_{t-2T}(i-T,j-T)\\ &\quad\oplus\psi_{t-2T}(i,j)\oplus\psi_{t-2T}(i,j-2T)\\ &= \psi_{t-2T}(i+2T,j)\oplus\psi_{t-2T}(i-2T,j)\oplus\psi_{t-2T}(i,j+2T)\\ &\quad\oplus\psi_{t-2T}(i,j-2T) \qquad (1.5)\end{aligned}$$

This result shows that property 1.4 is then also true for $2T$. Therefore, it is true for any lag T which is a power of two. At these particular values of time, the $\oplus$ rule is equivalent to the superposition (in the sense of the addition modulo two) of the initial pattern translated in the four lattice directions by an amount T. When the spatial extension of the initial pattern is small enough, we see it replicated four times. Otherwise destructive interferences show up, which give rise to partial replication.

For a square lattice of size $L = 2^k$, relation 1.4 implies that, after $L/2$ iterations, the result of the $\oplus$ rule is to superpose the initial condition with itself and, therefore, to yield a zero configuration.

For a number of iterations that is not a power of two, some results can also be obtained. In general, the configuration after T steps is the superposition modulo 2 of 4^k different translations of the initial pattern, where k is the number of digits equal to 1 in the binary representation of T.

This property can be proved as follows. First, we notice that the rule is additive, that is any initial pattern can be decomposed into the superposition of one-pixel images. Each of these simple configurations can be evolved independently and the results superposed to obtained the final complete pattern.

We will prove our statement for an initial configuration such that $\psi_0(0,0) = 1$, only. It is convenient to decompose T as a sum of powers of two. There exists a value of n such that we can write T as

$$T = 2^n + \sum_{\ell=0}^{n-1} a_\ell 2^\ell = 2^n + T' \qquad (1.6)$$

where, by construction, $T' \leq 2^n - 1$ (equality holds when all the $a_\ell = 1$).

Now, performing T iterations of the rule is equivalent to first do T' steps and then the last 2^n iterations. Clearly, by the definition of the rule, $\psi_{T'}(i,j)$ will contain the terms $\psi_0(i-T',j)$, $\psi_0(i+T',j)$, $\psi_0(i,j-T')$ and $\psi_0(i,j+T')$ and other terms involving only translations by a *smaller* amount. Therefore, the configuration at time T' has a given spatial extension.

When performing the next 2^n iterations, we know from 1.4 that they will result in the superposition of four translations by 2^n of the configuration at time T'. None of these translations will give rise to cancellation because the spatial extension of the configuration at time T' is smaller than the translation length. Indeed, after the left translation by 2^n, the right boundary will move to position $T' - 2^n$ with respect to the original pattern. Similarly, the left boundary will move to $-T' + 2^n$, due to the right translation. There is no overlap between the patterns generated in this way because, since $T' < 2^n$, one has

$$T' - 2^n \leq -T' + 2^n \tag{1.7}$$

Therefore, for each non-zero a_ℓ in expression (1.6), four translations are produced and the final result is composed of 4^k non-over lapping translations, with $k = \sum a_\ell$. When the initial image is not a single pixel, destructive interference is observed.

As a result, we get the rich geometrical structure observed in figure 1.2. As the number of iterations increases more and more terms are generated. Therefore, the algorithmic complexity of the expression becomes larger, thus reflecting the complexity of the cellular automaton configuration. More precisely, k is bounded by the logarithm of T. In order to evaluate the asymptotic complexity of the expression, we write $T \sim 2^k$. The number of translations generated is $4^k = T^2$ and the complexity goes as the square of the number of iterations.

The above discussion has unraveled the mechanisms leading to the complex structures produced by the $\oplus$ rule as being due to the superposition of the initial pattern translated many times by a different amount. From this analysis, we can conclude that this rule is not a self-replicating cellular automaton in the sense imagined by von Neumann.

Finally, it is interesting to note that the $\oplus$ rule is a generalization in two dimensions of the famous rule 90 of Wolfram which will be discusseded in section 2.1.1.

1.3 Definitions

1.3.1 *Cellular automata*

In this section we shall present a more formal definition of a cellular automaton. In general, a cellular automaton requires