Laboratoire de l'Informatique du Parallélisme

École Normale Supérieure de Lyon
Unité de recherche associée au CNRS n° 1398

An introduction to Cellular Automata

Marianne Delorme

July 1998

Research Report N° 98-37

École Normale Supérieure de Lyon
46 Allée d'Italie, 69364 Lyon Cedex 07, France
Téléphone : +33(0)4.72.72.80.00
Télécopieur : +33(0)4.72.72.80.80
Adresse électronique : `lip@ens-lyon.fr`

An introduction to Cellular Automata

Marianne Delorme

July 1998

Abstract

We give basic definitions necessary to understand what are cellular automata, as well as to work with. Some efficient but sometimes problematic concepts as signal, simulation and universality, are pointed out. In particular, different notions of universality are put to light.

Keywords: Cellular automata, universality, signals, simulations.

Résumé

On donne ici les définitions essentielles pour comprendre les automates cellulaires et les utiliser. On met en évidence certaines notions très éfficaces mais parfois problématiques, comme celles de signal, simulation et universalité. En particulier on présente différentes notions d'universalité.

Mots-clés: Automates cellulaires,universalité,signaux, simulations.

An introduction to Cellular Automata*

Marianne Delorme

22nd July 1998

1 Introduction

At the beginning of this story is John von Neumann. As far back as 1948 he introduced the idea of a *theory of automata* in a conference at the Hixon Symposium, September 1948 [77]. From that time on, he worked to what he described himself not as a theory, but as "an imperfectly articulated and hardly formalized "body of experience"" (introduction to "The Computer and the Brain", written around 1955-56 and published after his death [78]). He worked up to conceive the first cellular automaton (he is also said to have introduced the *cellular* epithet [10]). He also left interesting views about implied mathematics, including logics, probabilities, leading from the discrete to the continuous [77, 80, 79].

It is not the place to write the fascinating history nor to point out the role, in the development of the occidental civilization, of the dream, idea, concept, realizations of (automata) machines or games. Let us only recall the cornerstones that are the works of Lulle, Leibnitz, Pascal, Descartes, Vaucanson, Babbage ... But, in 1948, the minds were under the influence of the Turing (universal) machines [76], the first neural networks [55] and also the natural and artificial automata of cybernetics [82] (an enthusiastic review of which von Neumann published in "Physics Today" [60]).

Through some of the different texts he left as [77, 78, 80, 79], it is clear that von Neumann deeply gave his attention to the comparisons between *natural automata*, as the brain or other adaptable or evolutive systems, and *artificial automata*, as, especially, the computers (at that time just constructed). He was very interested in their respective complexities, which both, but in different ways, come under analog and digital procedures. The main points

*This report will appear as a contribution in the book *Cellular Automata: a parallel model.*, M. Delorme and J.Mazoyer Eds, Mathematics and Its Applications, Kluwer at issue were, on the one hand, the reliability of the computers, in danger to fail under increasing complexity, and, on the other hand, the power of the complexity of living organisms which seems to be a condition for their ability to reproduction. Thus von Neumann looked for an artificial system which would be as powerful as a universal Turing machine (that meant be *universal for computation*), would be able of reproduction and, even more, to construct, in a sense which has to be and will be made clear later, some significant inner components (that meant be *universal for the construction*).

J. von Neumann did not succeed to build the first model in three dimensions he tried, called the *cinematic model* by Burks and described in [10]. Then, on a suggestion of S. Ulam, he made up his mind to conceive a "2-dimensional paper automaton", with a *cellular* or *crystalline* structure, and he managed to get a cellular automaton, universal for computation and construction, and also self-reproducing. Actually the device was completed by Burks [10], afterwards improved by a lot of people [75], [16] for example, and it has had a rich posterity, as the following of the volume will partly show it. Actually, two large branches will expand. Along the first one, cellular automata are studied as *parallel models of computation* and *dynamic systems*. That leads to algorithms setting-up, language or pattern recognition, complexity theory, classifications ... Along the second one, cellular automata are conceived as *models for natural processes* in physics, chemistry, biology, economy ... Two purposes are pursued in this modeling process: to simulate phenomena on cellular automata, but also to try to predict phenomena in studying properties of relevant cellular models.

Now we will give basic definitions necessary to understand what are cellular automata, as well as to work with. But we also want to point out some efficient, and sometimes problematic, concepts as *signal*, *simulation* and *universality*. We will, especially, put to light different notions of universality which emerge more or less explicitly with time.

2 Cellular automata: main definitions

Informally, a cellular automaton or cellular space is an abstract object, with two intrinsically tied components. First a *regular*, *discrete*, *infinite* network, which represents the *architecture*, the *universe* or the *underlying space structure* of the cellular automaton, depending the use it will be made of it. Second, a finite automaton, a copy of which will take place at each node of the net. Each so decorated node will be called a *cell* and will communicate with a finite number of other cells, which determine its *neighborhood*, geometrically uniform. This communication, which is *local*, *deterministic*, *uniform* and *synchronous* determines a *global evolution* of the system, along *discrete time steps.*

2.1 Classic cellular automata

We will first give the standard formal definition, then indicate some more or less usual variants.

Definition 1.
A d-dimensional cellular automaton (or d-CA), $\mathcal{A}$, is a 4-uplet $(\mathbb{Z}^d, S, N, \delta)$*, where :*

- *S is a finite set, the elements of which are the* states *of $\mathcal{A}$,*
- *N is a finite ordered subset of $\mathbb{Z}^d$, $N = \{\vec{n_j}/\vec{n_j} = (x_{1j}, \ldots, x_{dj}), j \in \{1, \ldots, n\}\}$, called the neighborhood of $\mathcal{A}$,*
- *$\delta : S^{n+1} \mapsto S$ is the local transition function or local rule of $\mathcal{A}$*[1]*.*

Among the states are, sometimes, distinguished states s, called *quiescent* states, such that $\delta(s, \ldots, s) = s$.

2.2 Neighborhoods

Let $\mathcal{A}$ be a cellular automaton $(\mathbb{Z}^d, S, N, \delta)$. The neighborhood of a cell c (including the cell itself or not, in accordance with convention) is the set of all the cells of the network which will locally determine the evolution of c. It is finite and geometrically uniform.

In principle, a neighborhood can be any ordered finite set, but, actually, some special ones are mainly considered. In case of $\mathbb{Z}^d$, classic neighborhoods are the von Neumann's and the Moore's ones. They are known as the *nearest neighbors neighborhoods*, and defined according to the usual norms and the associated distances. More precisely, if $\vec{z} = (z_1, \ldots, z_d)$, $\|\vec{z}\|_1$ will denote $\sum_{i=1}^{d} \mid z_i \mid$, $\|\vec{z}\|_\infty$ will denote $Max\{\mid z_i \mid \;\; / i \in \{1, \ldots, d\}\}$, d_1, d_∞ the associated distances, and we get:

- Von Neumann neighborhood: $N_{VN}(\vec{z}) = \{\vec{x}/\vec{x} \in \mathbb{Z}^d$ and $d_1(\vec{z}, \vec{x}) \leq 1\}$ with a given order,

[1]This definition takes into account the cell and its neighborhood. It is a quite common choice, but not a universal one: Cole [17], for example, does not consider the cell in its nighborhood.

- Moore neighborhood : $N_M(\vec{z}) = \{\vec{x}/\vec{x} \in \mathbb{Z}^d \text{ and } d_\infty(\vec{z},\vec{x}) \leq 1\}$, with a given order.

Some other interesting neighborhoods appear in the literature, the Cole's ones [17], or some others in [69] for example which are represented with the Von Neumann's and the Moore's ones on Figure 1, when the cells are considered on the vertices or, more usually, by duality, as the unit squares of the grid.

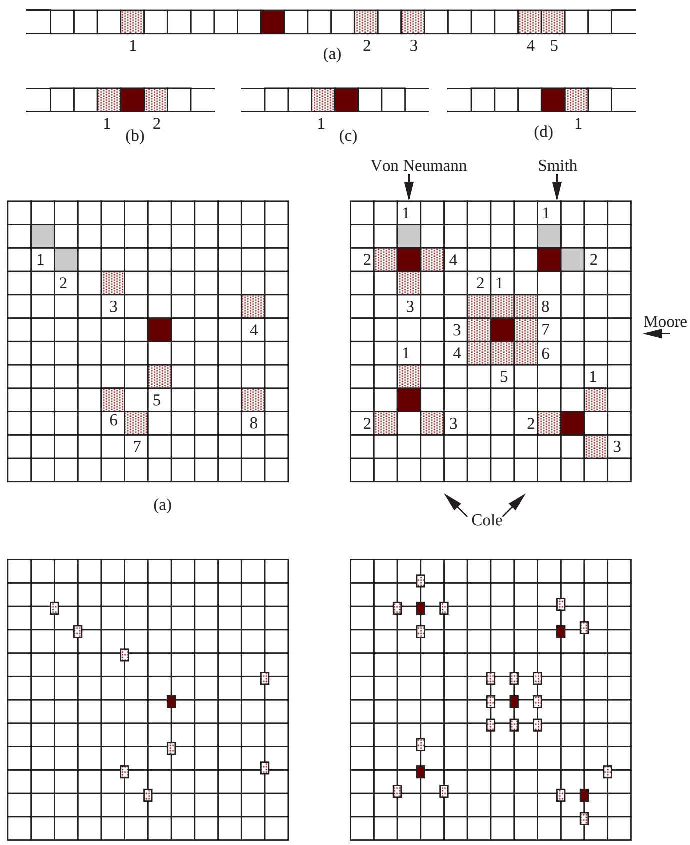


Figure 1: Examples of classic neighborhoods (the numbers mark the cells ranks). In dimension 1, (b) represents, in the two-way case, von Neumann's or Moore's neighborhood, which then coincide, while (c) and (d) represent the first neighbors neighborhood in case of one-way cellular automata. In both dimensions, (a) represents any neighborhood.

Moreover the following result holds, which means that, *from a computation point of view*, all neighborhoods can be understood as *equivalent*, which is generally not the case when complexity or architecture are to be taken into account (see for example [16], [69], [46]).

Fact 1.

Every d-cellular automaton can be simulated by a d-cellular automaton with the nearest neighbors neighborhood.

Another neighborhood on 1-cellular automaton is used which consists of the closest right (left) cell of a given cell. It characterizes cellular automata called *one way cellular automata* (OCA for short), which form a special interesting class (see [15], [40], [41], [74] and [73] for example).

The *radius* r of a neighborhood is defined as the distance between the cell of which it is the neighborhood and the farthest one in the neighborhood. So, for one-dimensional cellular automata, the radius of the nearest neighbors neighborhood is $r = 1$ and the number of cells is $2r + 1 = 3$. This terminology and the corresponding notation are often used in case of symmetric neighborhood in dimension 1.

The neighborhoods for which the ratio between the radius and the number of states is exponential are not considered in the complexity domain, because, in this framework, NP-complete problems may be solved in polynomial time. Nevertheless *an interesting problematics is the optimization of the constraint "size of the neighborhood/number of states".*

2.3 Representations

Outside the usual transition table or matrix, the local transition function can be displayed in different and more efficient ways. The first one, due to Wolfram, is well known and refers to the rules of its, now historical, classification.

Representation 1. *Wolfram numbers*

We can describe δ by a word obtained in the following way: first of all a linear order on the elements of the δ-domain S^{n+1}, interpreted as words of length $n+1$ on S, is chosen. Then δ is given by a word the letters of which are the successive δ-values of the ordered words of the δ-domain.
When working with 1-dimensional cellular automata, $S = \{0,1\}$ and the lexicographic order on $\{0,1\}^{n+1}$, we get a word which can be understood as the binary development of a positive integer. This integer is the Wolfram number of the rule δ.

So, for 1-dimensional cellular automata with $\{-1,0,1\}$ neighborhood (cell included in the middle), $\{0,1\}$ set of states, the lexicographic order $\{w_0, \dots, w_7,\}$ on $\{0,1\}^3$, if $\delta(w_i) = s_i$, $s_0 \dots s_7$ represents δ. And conversely, any positive integer less than $256 = 2^8$ defines a 1-CA of the last type. For example, as $54 = 2^1 + 2^2 + 2^4 + 2^5$, the "54 rule" is given by the table below:

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

Another representation is very fruitful for 1-CA with nearest neighborhood N, $|N| \geq 3$. It is connected to the idea to link a 1-CA $(\mathbb{Z}, S, N, \delta)$ to a finite automaton with S^n as set of states, letters in S, and the transition function δ' defined by $\delta'(su, \delta(sut)) = ut$ for $s, t \in S$, $u \in S^{n-1}$, the underlying graph of which is a De Bruijn graph. The properties of De Bruijn graphs (as, for example, being connected and hamiltonian) can be successfully exploited to get decidability results on CA as in [36], [71]. See also [72].

Representation 2. *De Bruijn graphs*

In the case of dimension 1*, we also get description of* δ *aid of a graph defined as follows: its vertices correspond to, and are labeled by, the words of length* n *on* S*, and there is an arrow from* su *to* ut*, labeled by* $\delta(sut)$*.*

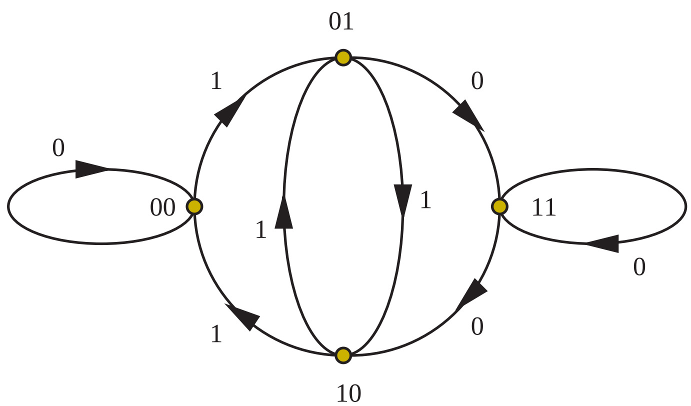


Figure 2: The De Bruijn representation of the "54 rule", $r = 1$

The above representations do not answer any concern for optimization. Some results can be found about minimization of the transition tables in [27].

2.4 Configurations and global function

The space component already comes to light with the neighborhood, but it takes its full significance with the notions of configurations and global function. We will first precise definitions and then recall some results.

2.4.1 Configurations, behavior, global function, dynamic system

If we conceive a cellular automaton $\mathcal{A} = (\mathbb{Z}^d, S, N, \delta)$ as a machine which evolves with time, knowing it at time t is knowing the state of each cell at

this time. This is formalized through an application $c_t^{\mathcal{A}} : \mathbb{Z}^d \mapsto S$, called a *configuration*, or *instantaneous description* or *global state* of $\mathcal{A}$ at time t.
The *behavior* or *action* or *evolution* of $\mathcal{A}$ is a sequence $(c_t^{\mathcal{A}})_{t\geq 0}$ of configurations determined by the given initial configuration $c_0^{\mathcal{A}}$ and the derivation rule described as follows:

$$c_t^{\mathcal{A}} \vdash c_{t+1}^{\mathcal{A}} \text{ if and only if } c_{t+1}^{\mathcal{A}}(\vec{z}) = c_{t+1}^{\mathcal{A}}(z_1, \ldots, z_d) \text{ is given by}$$
$$\delta(c_t^{\mathcal{A}}(z_1, \ldots, z_d), c_t^{\mathcal{A}}(z_1 + n_{11}, \ldots, z_d + n_{d1}), \ldots, c_t^{\mathcal{A}}(z_1 + n_{1n}, \ldots, z_d + n_{dn})),$$

where, $\vec{z}$ represents the standard cell and $(\vec{n}_1, \ldots, \vec{n}_n)$ the neighborhood. This relation actually gives rise to an application $G_{\mathcal{A}}$ from the set $\mathcal{C}_{\mathcal{A}}$ of $\mathcal{A}$-configurations into itself, which associates to each configuration $c^{\mathcal{A}}$ the configuration $G_{\mathcal{A}}(c^{\mathcal{A}})$, defined by :

$$G_{\mathcal{A}}(c^{\mathcal{A}})(\vec{z}) = \delta(c^{\mathcal{A}}(\vec{z}), c^{\mathcal{A}}(\vec{z} + \vec{n}_1), \ldots, c^{\mathcal{A}}(\vec{z} + \vec{n}_n)).$$

The function $G_{\mathcal{A}}$ so obtained is the *global function* of $\mathcal{A}$. It plays an essential role in practicing cellular automata, especially when they are considered as dynamic system. Indeed $(\mathbb{Z}^d, G_{\mathcal{A}})$ is a dynamic system[2], but a very special one due to its *local* definition via the local transition function and to its homogeneity.

A sequence $(c_t^{\mathcal{A}})_{t\geq 0}$ of configurations of a cellular automaton $\mathcal{A}$ is designated, according to the context, as the *orbit* of $c_0^{\mathcal{A}}$ when cellular automata are considered as dynamic systems, or as a *computation* on $c_0^{\mathcal{A}}$ when cellular automata are seen as computation models or even as a *propagation* or a *positive motion* in a more physical point of view.
In the computation case, each derivation $c_t^{\mathcal{A}} \vdash c_{t+1}^{\mathcal{A}}$ is called a *computation step*. Their evaluation determines the computation time of a computation or the time complexity of a problem or a language.

Incidentally, we have to notice that the notion of computation does not distinguish *computation time* and *communication time* between neighbor cells. This does not pose problem here because our aim is more to exhibit parallel algorithms than to conceive physical or "real" machines.

2.4.2 Special configurations

Cellular automata are not "real" machines, in the sense that they don't exist physically but have to be simulated. So, when experiment has to be done or some representations of their evolutions are to be given, some types of *feasible or significant* configurations are of special interest, which also are

[2] $G_{\mathcal{A}}$ is then often called the evolution or next state transformation, configurations being the states of the system.

amenable for mathematical treatment. Among them, the following ones play essential roles.

- *Finite configurations*
  Let q_e be a quiescent state for some cellular automaton $\mathcal{A}$, and c a configuration. The *support* of c is the set $Supp(c) = \{\vec{z}/c(\vec{z}) \neq q_e\}$. Then a *finite configuration* is a configuration the support of which is finite.
  As data are there finite, these configurations are essential in the computation or language recognition areas, and so play a basic role in computation, computation universality and complexity.
  Let us remark that when several quiescent states exist, which is often encountered, one of them is explicitly reserved for the definition of finite configurations.

- *Periodic configurations*
  A *periodic configuration* is a configuration c for which there exists $\vec{p} \in \mathbb{Z}^d$ such that, $c(\vec{z}+\vec{p}) = c(\vec{z})$ for each $\vec{z} \in \mathbb{Z}^d$.
  Note that, except the *quiescent configuration* c_{q_e} (such that $c_{q_e}(\vec{z}) = q_e$ for each $\vec{z} \in Z^d$) periodic configurations are not finite. They are worthy to study because they naturally come for automata on rings (which are those we are able to "observe" on computers) and more generally n-dimensional tori, but also because they constitute "good" subsets of $S^{\mathbb{Z}^d}$, in fact countable dense ones (as well as the finite ones) for the canonical topology recalled in the next section.

  Let us here pass some remark on the terminology. Actually, we should name *space-periodic* the above periodic configurations, in order to distinguish them from *time-periodic* configurations which are these configurations c the orbits of which are periodic, which means that there exists a time t_0 and a positive integer p such that, for all positive integers k, $G_{\mathcal{A}}^{t_0+kp}(c) = G_{\mathcal{A}}^{t_0}(c)$, where $G_{\mathcal{A}}^{t}(c)$ denotes the result of the t-th iteration of $G_{\mathcal{A}}$ on c. This last notion is very natural when cellular automata are considered as dynamic systems. Clearly space periodic configurations are also time-periodic, but the converse is, generally, not true.

- *Garden of Eden*
  The so called configurations are those which have no predecessor according to the global function. So they can only be taken as initial configurations (which explains the terminology), and, consequently, only "given" by external input. They appeared in the context of construc-

tion universality and self-reproduction, and, in fact, originated the systematic study of global functions [59], [61] and [20].

2.4.3 Some results and open problems

Let us consider a finite set S, a positive integer d, $d \geq 1$. If S is endowed with the discrete topology (for which all subsets are open), the set $S^{\mathbb{Z}^d}$ of applications from $\mathbb{Z}^d$ into S, canonically endowed with the product topology, is a compact metric space. A *shift of vector* $\vec{v}$, $\vec{v} \in \mathbb{Z}^d$, is an application $\sigma_{\vec{v}}$ from $S^{\mathbb{Z}^d}$ into itself, such that, for each c, $c \in S^{\mathbb{Z}^d}$, each $\vec{z}$, $\vec{z} \in \mathbb{Z}$, $\sigma_{\vec{v}}(c)(\vec{z}+\vec{v}) = c(\vec{z})$. Then one of the source of interest in cellular automata can be found in the basic following result, proved independently of the same one (but in case $d = 1$) by Hedlund [37].

Theorem 1. *Richardson's theorem, [65]*
An application $f : S^{\mathbb{Z}^d} \longrightarrow S^{\mathbb{Z}^d}$ which commutes with shifts is a continuous function if and only if it is the global function of some cellular automaton.

Some classical properties of the global functions, as surjectivity, injectivity and bijectivity, have been intensively studied. A first result is the Moore-Myhill theorem, often known as the Garden-of-Eden theorem.

Theorem 2. *Garden-of-Eden theorem, [59], [61]*
A cellular automaton with a quiescent state is surjective if and only if it is injective when restricted to finite configurations.

Another property of cellular automata has been intensively studied, from different areas but especially when looking cellular automata as models of physical phenomena (and particularly in microscopic physics where processes can be reversible), the *reversibility.* A cellular automaton is said to be *reversible* when its global function G is bijective and G^{-1} is also realized by a cellular automaton.
A consequence of Richardson's and Moore-Myhill's theorems is that *bijectivity, reversibility and injectivity* are equivalent for cellular automata.

We can now sum up the main other results in Figure 3, taken from [27] where they are reviewed and/or proved.

Notice that in the above cited paper are also mentioned decidability and undecidability results, which are to be stressed. First they reveal the existing gap between 1- and 2-dimensional cellular automata: actually many problems are decidable in dimension 1 while they are undecidable in dimension 2. Secondly, the undecidability proofs, involved enough, put to light tight links between cellular automata and tilings, and so exhibit another powerful proof

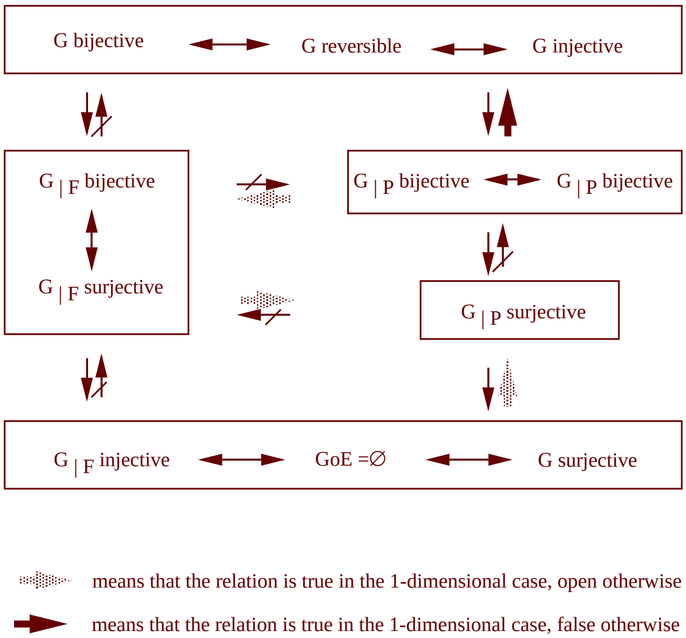


Figure 3: $G|_F$ denotes the functions restricted to finite configurations, $G|_P$ those restricted to the periodic ones and GoE the Garden-of-Eden configurations.

method, to be opposed to the diagonalization-like methods. Let us simply set out the basic result by Kari, which is exemplary.

Theorem 3. *Kari's theorem [44], [27]*
The surjectivity and reversibility problems are undecidable for 2*-dimensional cellular automata.*

We will now pay more attention to the configurations evolutions.

2.5 Dynamics

2.5.1 Space-time diagrams

A cellular automaton deals with information. It works on given information, converts, creates, cancels, stores and delivers some. It can be very useful to represent all these transformations which occur along discrete times. For cellular automata of low dimension, especially 1 and, to lesser extent, 2, we get very expressive graphical representations, their *space-time diagrams* in $\mathbb{N} \times \mathbb{Z}^d$, obtained in attributing a special pattern or color to each state, and consequently to each cell of the automaton. Thus, the evolution of the automaton on a given starting configuration (also designated as initial configuration) can be seen as a stacking-up of decorated lines in $\mathbb{Z}^2$, or of decorated planes in $\mathbb{Z}^3$ depending on whether the automaton dimension is 1 (see Figure 4) or 2.

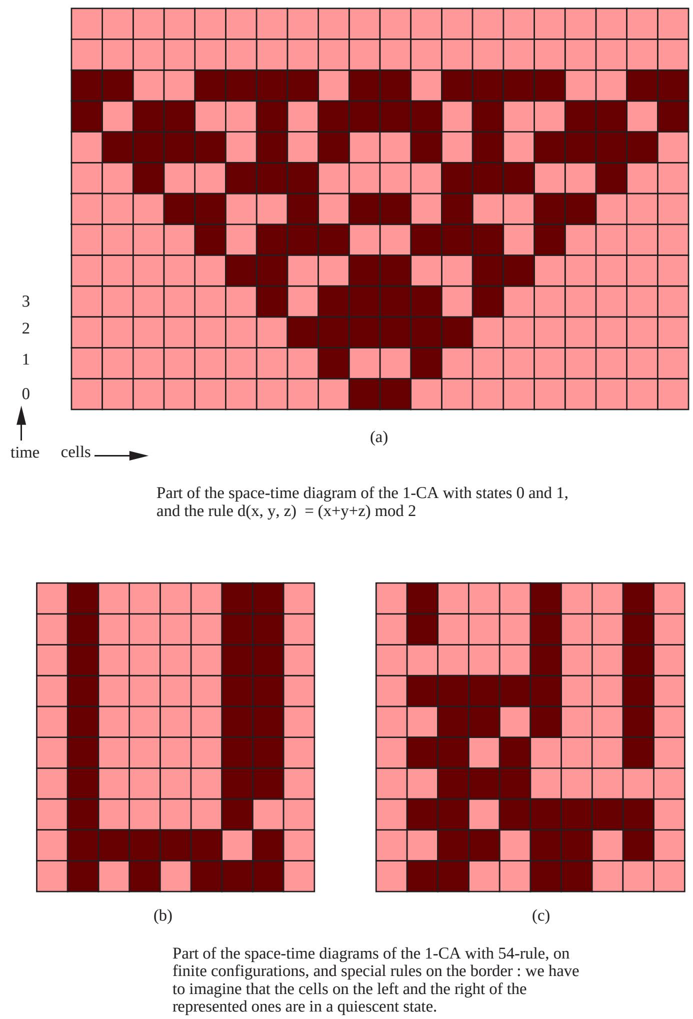


Figure 4: Example of space-time diagrams on finite configurations. On (a) the computation is unbounded while it is bounded on (b) and (c).

These examples, although very simple, put already to light two possible kinds of computations : bounded and unbounded ones. See for example [23] and [24].

It is quite easy to imagine that graphical representations are suffering from technology limits (in space - most simulations are done on rings of cells - and time), nevertheless they are powerful and rich enough to make understand or imagine phenomena and algorithms open to cellular automata treatments.

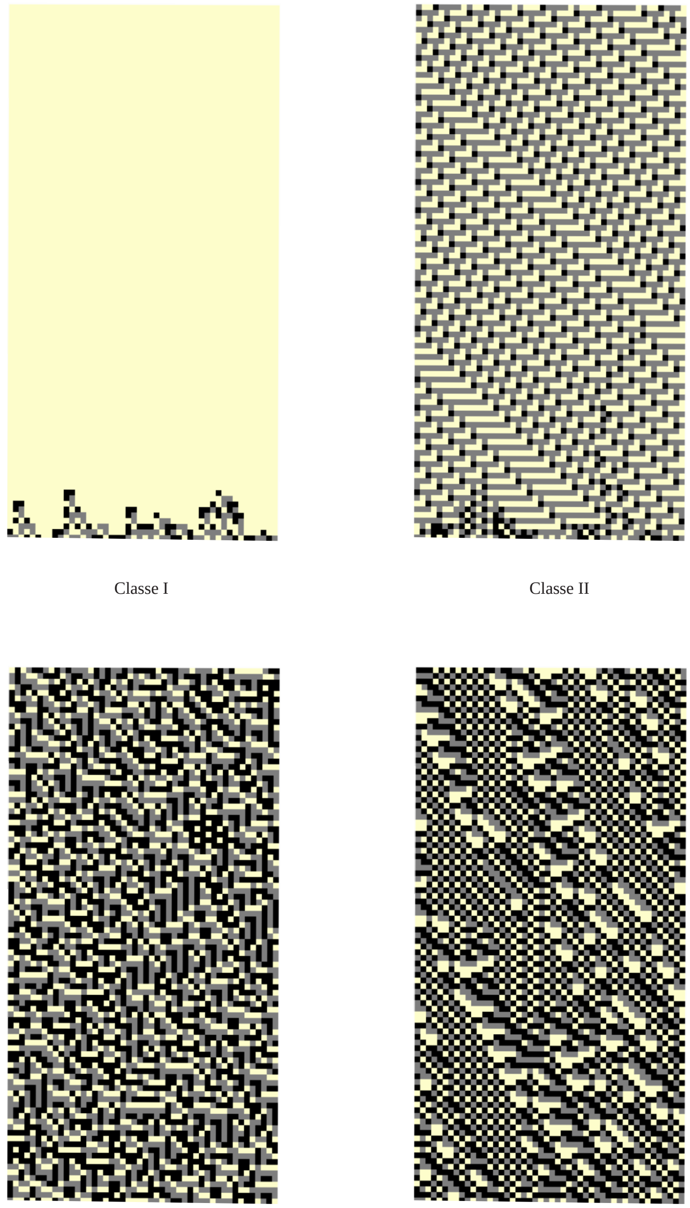
Outside the production of attractive or strange images, and the question of the connections between cellular automata and fractals that were naturally set up ([83], [21], more recently [62] and references there), the observation and analysis of the complexity, or degree of intricacy, of such space-time diagrams for 1-dimensional cellular automata is at the foundation of some classifications for these automata. The first and widespread one by Wolfram [85], focuses on the behavior of 1-dimensional cellular automata with two or three states on periodic configurations, and rests on the "observation" of cellular automata evolutions on random initial configurations. Unfortunately, these space-time diagrams observations don't allow any formal definition of the evolution complexity, so the corresponding heuristic classification remains coarse even if it is suggestive enough and, especially, gives evidence of some "edge", sometimes evoked as "the edge of chaos", or of a possible computation-universality for some automata. See Figure 5.

The study of these diagrams has also inspired concepts and methods efficiently used in cellular automata investigations, as, for example, the ones of signal and grouping. We will later come back to these notions, but let us already note that an attempt of algebraic classification, founded on a grouping notion, could give a new look on the complex world of cellular automata [53]. Let us also remark that practicing these diagrams has also certainly lead to the idea of a better use of some "inactive" computational areas as, for example, in computations developed in [52].

2.5.2 Phase space

Another way to analyze the global function of a cellular automaton $\mathcal{A}$ is to structure the configurations set into a graph according to the above derivation relation. More precisely the following graph, called the *phase space* of $\mathcal{A}$, is considered: its vertices are the configurations, and there is an arc from c to c' if and only if $c' = G_{\mathcal{A}}(c)$.
Actually in this space we get two sorts of orbits. Those corresponding to time-periodic configurations c which split into two parts : a first one such that $G^t_{\mathcal{A}}(c) \neq G^{t'}_{\mathcal{A}}(c)$ for all $t' < t$, called the *transient* part of c, and a second one which is a cycle, sometimes called the *period* of c. And those corresponding to orbits without cycle. So the phase space appears as a set of graphs (a forest), which obviously are the orbits of the Garden-of-Eden configurations of the automaton!

It is generally not possible to completely handle the phase space of a cellular automaton. Nevertheless, for *additive* or *linear* cellular automata, the phase space restricted to the periodic configurations can be completely de-



Classe I

Classe II

Classe III

Classe IV

Figure 5: Other examples of space-time diagrams, characteristic of the four Wolfram classes.

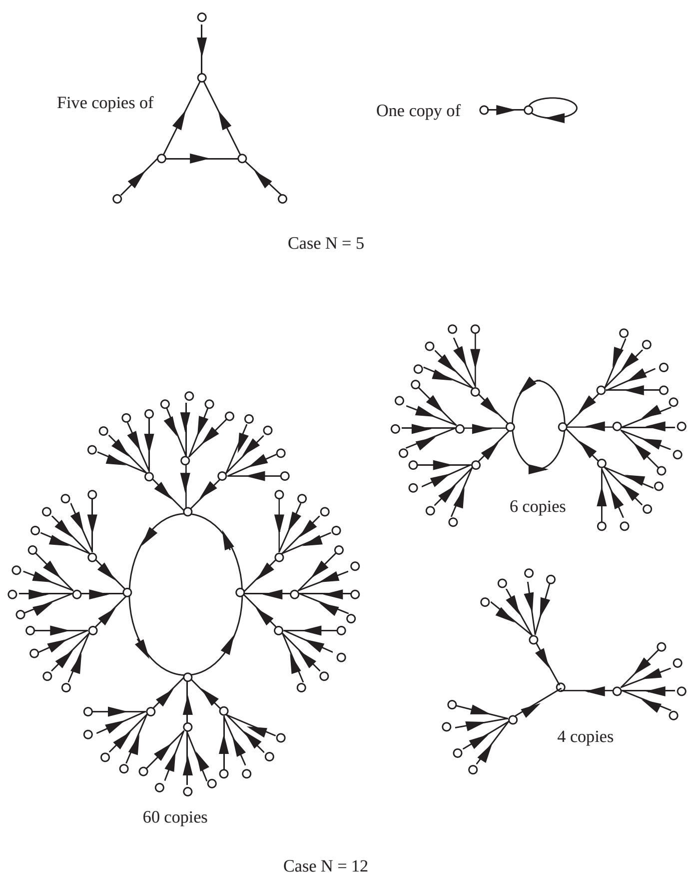


Figure 6: Phase space of the cellular automata defined on N-cells rings, the configuration of which are given, at time t, by $c^{(t)}(x) = \sum_{j=0}^{N-1} a_j^{(t)} x^j$ with $a_j^{(t)} = a_{j-1}^{(t-1)} + a_{j+1}^{(t-1)}$ and $a_j^{(t)} \in Z/2Z$.

scribed via some polynomials canonically associated to the global functions [48], as it is the case for the examples of Figure 6. Incidentally, let us note here that additive cellular automata have been much studied : their "arithmetic" nature allows to think them easier to understand although they may have arbitrary complex behaviors (along with [48], see [4], [43], [71], [2], [3] and relevant references in these papers). Some attempts to generalize linear cellular automata are set about, with bilinear ones for example [6], [7].

The above mentioned difficulty leads to pay attention to the canonical topological structure (evoked in 2.4.3.) of the configurations set (then also named phase space) and especially to some of its subsets considered as giving some good approximation of the global asymptotic cellular automaton behaviour on some configurations. Among them appears *the limit set*, introduced by Wolfram [85], which represents the set of configurations such that their predecessors set (for the global function) is a tree of infinite depth.

Definition 2. *Limit set*

Let $\mathcal{A} = (S, \delta)$ be a d-cellular automaton, and $G_{\mathcal{A}}$ its global function. The limit set of $\mathcal{A}$ is $\Omega(\mathcal{A}) = \cap_{t \geq 0} \Omega^t$ with $\Omega^0 = S^{\mathbb{Z}^d}$ and $\Omega^t = G_{\mathcal{A}}(\Omega^{t-1})$ for $t \geq 0$.

This notion (which makes sense because $\Omega(\mathcal{A})$ is non empty, due to the fact that the space is a compact one) allows to separate cellular automata into two classes: class 1 contains all cellular automata for which there exists i such that $\Omega^i = \Omega$ and class 2 the others, which is made relevant by the following result proved in [20][3]: *$\mathcal{A}$ belongs to class 2 if and only if there exists a countable intersection D of dense open sets such that $\Omega \cap (\cup_{i \in \mathcal{N}} G^i_{\mathcal{A}}(D)) = \emptyset$.* Most non trivial properties of the limit sets are recursively undecidable, even when the sets are restricted to finite configurations (see [20]).

When we regard a configuration as a bi-infinite word on the finite alphabet S, we can consider its finite factors. The finite factors of the elements of $\Omega(\mathcal{A})$, for 1-CA's $\mathcal{A}$, make up a language, *the limit language* of $\mathcal{A}$, that has been intensively studied. It was proved that there exist 1-cellular automata with regular, context-free, context sensitive, recursive-enumerable and non-recursive-enumerable limit languages ([39], [20], [35]), all results that contribute to answer a question set up by Wolfram (Problem 13 in [84]). Let us remark that reversible cellular automata have rational limit sets: the whole set of words. As there exist universal reversible cellular automata ([58], [26]), there exist universal cellular automata with rational limit sets. But these sets are trivial. It is not always the case: in [35] a cellular automaton is built, which is universal and has an *explicit* non trivial rational limit language.

[3]Another classification, may be the first formal one, was established in [19] for special finite configurations. Later, S. Ishii in [42], proposed a new classification based on a dynamic property of orbits of "almost every" configurations.

2.5.3 The work in progress

Problems studied in the dynamics field are mainly of two sorts. On the one hand, general, mathematical properties of the global function are investigated, as surjectivity, injectivity, bijectivity and shift invariance, as it was already mentioned. On the other hand, topologies or metrics are defined on the phase space and their properties checked, especially those which seem to be relevant to found significant classifications. See [13] and [9]. All these authors study the phase-space, but their approaches are quite different. People around Cattaneo start from cellular automata (so from the dialectic local/global) and look, among other investigations, for a notion of chaos which better corresponds to some "cellular" intuition, while people around Kůrka apply results on dynamic systems to the particular ones that are cellular automata.

To finish, let us emphasize that, in this area, contrary to what happens in the computation one, all configurations are taken into account, especially infinite ones.

2.6 Various types of cellular automata

Some types of cellular automata are of special interest, on various argument.

- *Totalistic cellular automata* play a particular role. They are 1-dimensional cellular automata (S, δ) such that there exists some function $f : \mathbb{N} \longrightarrow \mathbb{N}$ defining δ, that is more precisely for automata with the first neighbors neighborhood : $\delta(s_1, s_2, s_3) = f(s_1 + s_2 + s_3)$.

- We have already mentioned *linear cellular automata*. One of their main property is to fulfil the *superposition principle*, which means that their global function is an endomorphism. In this class we find, in particular, cellular automata the space-time diagrams of which are superposition of Pascal's triangles modulo integers (see [63]).

- Other types of cellular automata are determined by restrictions due to their functions, as languages recognizers for example. In this latter case the input mode also lead to different types of automata as can be seen in [24].

2.7 Cellular automata on Cayley graphs

A natural generalization of cellular automata has been *graphs of automata*, obtained from undirected graphs (finite or infinite, but of finite degree), in putting at each vertex of the graph a copy of one finite automaton, the graph edges defining the connections between the automata. But, such a radical generalization gives birth to difficulties, some of which are discussed in [64]. In particular the regularity of the graph as the uniformity of the neighborhood are lost, and a cell does not know its environment. A way to avoid it is to narrow the definition to a special class of graphs, the Cayley graphs. Cayley graphs were introduced by Cayley [14] for "drawing groups". His research for representing groups originated a lot of beautiful problems and results in group theory, and in studying cellular automata on Cayley graphs we can take advantage of it. Moreover, we know that each finite graph can be regarded as an induced subgraph of a Cayley graph (*Let Y be a graph with n vertices and let G be a group. Then some Cayley graph of G has Y as an induced subgraph if* $|G| \geq (2+\sqrt{3})n^3 \approx 3.74n^3$ [34]). Finally, as Z^d can be interpreted as a Cayley graph, we also get with cellular automata on Cayley graphs a direct and natural generalization of standard cellular automata.

But the idea of considering cellular automata on Cayley graphs can also be justified "from the outside". Some modelizations, in Physics or in Biology, use different tilings of their spaces, as tilings by triangles or hexagons for example, and people thought it was interesting to study the power of cellular automata on such structures. It turned out to be of low interest as shows the Róka's result just below. But other tilings remain to be studied.

Some work has already been done on these objects [47], [67]. In the former one the Moore-Myhill theorem is extended to automata on Cayley graphs of groups of non exponential growth.

Theorem 4. *[47]*

Let $\mathcal{G}$ be the Cayley graph of a group of non exponential growth. Then, for a cellular automaton on $\mathcal{G}$, there exist Garden-of-Eden configurations if and only if there exist two mutually erasable configurations[4].

We also will here emphasize the following result, which could be rephrased in the computer science frame: *every reasonable plane architecture is equivalent to the grid.*

Theorem 5. *[67]*

For some given notion of simulation, the cellular automata on the Cayley graphs that define any archimedian tilings of the plane are equivalent.

[4]This terminology was also the terminology of Moore and Myhill, which was gradually transformed with time into the one used in 2.

2.8 Other generalizations

If cellular automata seem to be very efficient as computation models, and if they give interesting results in the field of modeling, some of their features imply that they are not so well adapted to biological processes. If a very big number of identical cells is efficient for computing, the rigidity and even the regularity of the structure is far from the biological reality.
So we can imagine extensions of these automata to more malleable models. See [18] for example.

Moreover it is also possible to replace the finite automaton (S, δ) by other types of machines (as pushdown automata, Turing machines, ...).

3 Signals

When working with cellular automata, a very useful and familiar notion is the one of *signal*. It is a fundamental notion or tool, but, paradoxically, it seems difficult to get a satisfactory formalization of it, perhaps because it arises from an intricate mixture of different levels of understanding. Actually, we can imagine a signal as the virtual track of some specific information transmitted from some cells to some others, along a specific way, through the universe, if we understand the universe as the space-time diagram of some initial configuration.

In fact, we have essentially two ways to cope with cellular automata: we have either to build an automaton achieving some task, or to discover or interpret what an automaton is performing. Let us take the case of 1-dimensional cellular automata. In the first case we can try to imagine what would be the automaton space-time diagram, and to find some way to geometrically structure this space to set up the result, what gives birth to geometric diagrams. Some significant real lines of the plane appear that can then be called *signals*, and, when they are *constructible* by cellular automata, they can be transformed into *cellular automata signals*. In the second case, we can observe some distributions or patterns of specific states which seem to be significant and can be considered as making up *cellular automata signals*.

Let us be a little more precise.

3.1 One dimensional signals

We are here interested in conceiving 1-dimensional cellular automata.

3.1.1 Signals in geometric diagrams

Suppose we intend to design a cellular automaton which marks the integer powers of 2. The geometric diagram (Figure 7) put to light efficient straight lines (A with equation $x = 0$, Δ with equation $y = 3x$) or segments (parts of the straight lines $\vec{D_{2^n}}$ with equation $y = x + 2^n$ and $\vec{D'_{2^n}}$ with equation $y = -x + 2^n$ between A and Δ), which are then interpreted as signals in a dynamical understanding of the behaviour of a cellular automaton.

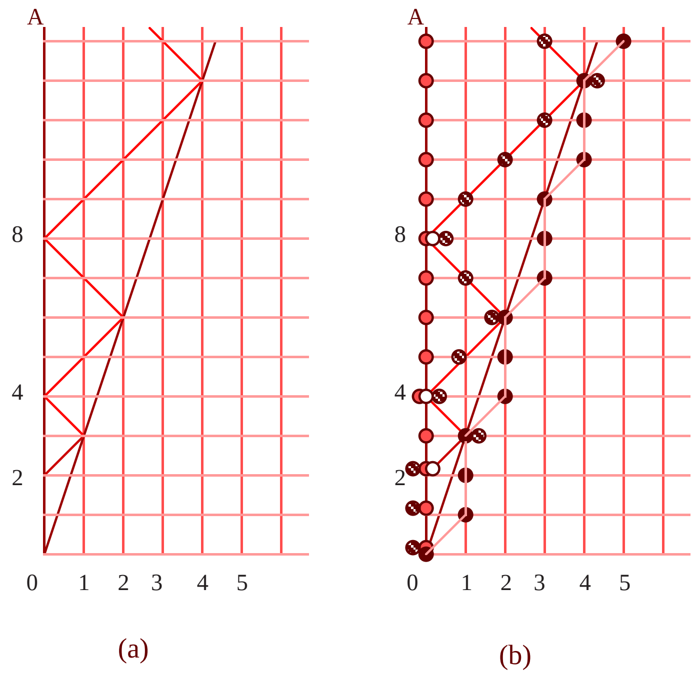


Figure 7: Geometric (a) and cellular (b) signals to obtain the powers of two on the cell 0.

And so it is very natural to say: to get the 2^n positions, $n \in \mathbb{N}$, on the cell number 0, it is sufficient to send, from this very cell, some *signals*: the first one will mark the cell 0, the second one will evolve as Δ, the third one will oscillate between A and Δ, at “maximal speed”, and will give a wanted integer each time it arrives on cell 0. Its meeting points with Δ are the points $(2^{n-1}, 3(2^{n-1}))$, $n \geq 1$ (see Figure 7(a)).

If we are going back to the line of cells, we can imagine signals as information handled as an indivisible unit, able to stay on cells or move between cells (as quickly as possible would be the best), and, sometimes, when meeting others, to modify their moves directions or, even, to give birth to new ones. These moves are controlled by means of local rules and finite (what is essential) sets of states.

Our example, given in [15], is simple, and it is not difficult to design a cellular automaton the behaviour of which allows to get the 2^n on the first cell, as shows the space-time diagram of the Figure 8.

3.1.2 Simple or basic cellular automata constructible signals

In the above example, we observe that signals on the geometric diagram are straight lines or segments of straight lines, which can be seen as the best trajectories of information between cells. Moreover, as time is not reversible, these lines are strictly increasing functions of it. Finally, as we are considering cellular automata with 1-neighborhood, the course of an atom of information will to be found inside a cone with the initial cell as vertex, so, modulo a possible rotation of 45°, we can limit our attention to sets of cells in $\mathbb{N} \times \mathbb{N}$, and the atom will stay on a cell or go to one of its consecutive neighbors. This leads to the first following definitions.

Definition 3. *Signals*

1. *A signal is a set* $\{(f(t), t)\}_{t \in \mathbb{N}}$*, where* f *is a function from* $\mathbb{N}$ *into* $\mathbb{N}$ *such that* $(f(t+1), t+1)$ *is* $(f(t)-1, t+1)$*,* $(f(t), t+1)$ *or* $(f(t)-1, t+1)$*. Such a signal is said to be rightward (resp. leftward) if, for each* t*,* $t \in \mathbb{N}$*,* $(f(t+1), t+1) \in \{(f(t), t+1), (f(t)+1, t+1)\}$ *(resp.* $\{(f(t), t+1), (f(t)-1, t+1)\}$*).*

2. *A signal is said to be simple or basic if the sequence* $(f(t+1) - f(t))_{t \in \mathbb{N}}$ *is ultimately periodic.*

We are interested in signals constructible by cellular automata. What does that mean?

Definition 4. *Constructible signals*

A signal Σ *is said to be constructible by cellular automata (or CA-constructible) if there exists a cellular automaton* $(S, \delta, \sharp, G, s_e, S_0)$ *where* $\sharp, G$ *et* s_e *are special states of* (S, δ)*,* S_0 *is a subset of* S*, such that*

- $\sharp$ *is the border state, considered as the unchanging cell-*0 *left neighbour,*

- s_e *is a quiescent state such that* $\delta(s_e, s_e, s_e) = \delta(\sharp, s_e, s_e) = s_e$*,*

- G *is a state such that, at initial time, all cells are in state* s_e *except the cell* 0 *that is in state* G*,*

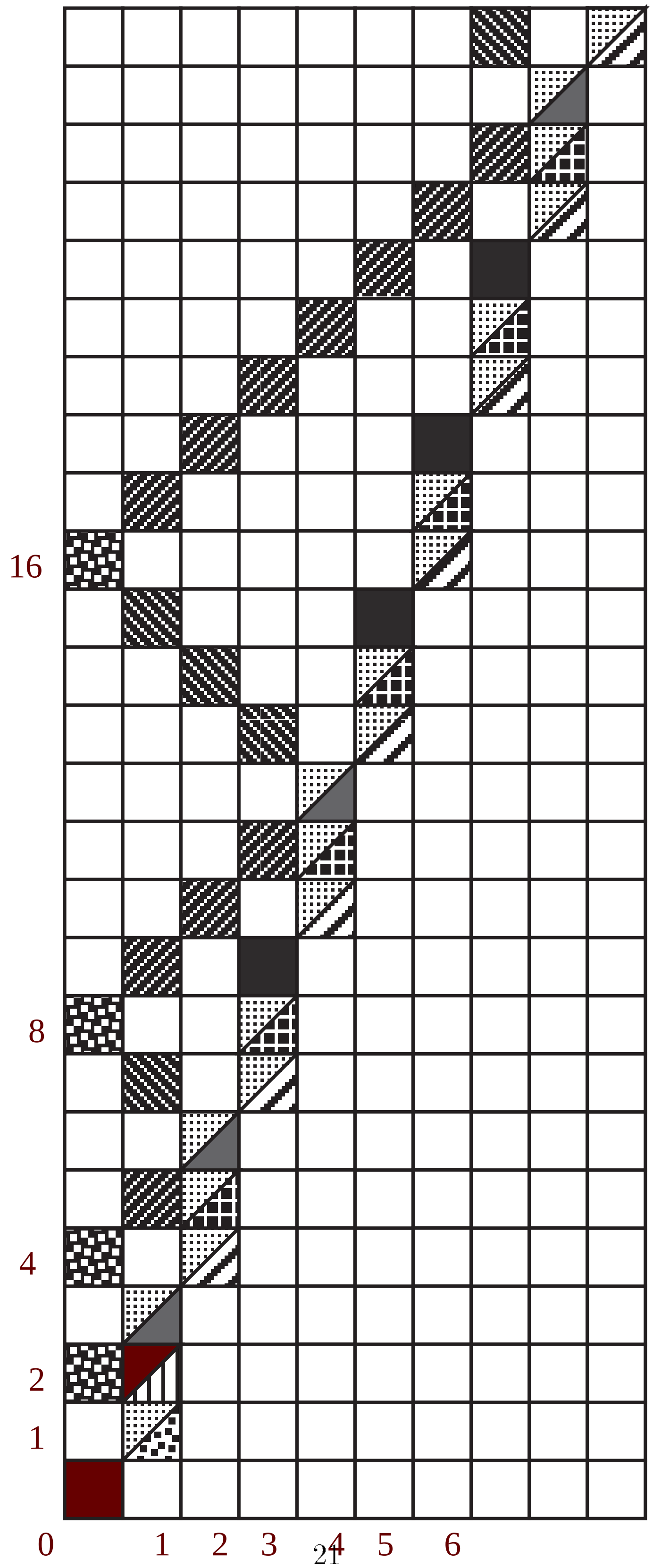


Figure 8: Space-time diagram for signals to obtain the powers of two on the cell 0.

- *a cell k, at time t, is in a state of S_0 if and only if (k,t) is in Σ.*

Then we have:

Fact 2.

Basic signals are CA-constructible. Moreover, if a signal Σ is generated by $(S, \delta, \sharp, G, s_e, S_0)$ in such a way that all sites outside Σ are in state s_e, then, Σ is finite or basic.

Let Σ be a basic signal, *per* its lowest period starting at t_0 and $\sigma = f(t+per) - f(t)$ for any t, $t > t_0$. The rational number $\frac{per}{\sigma}$ is called the slope of Σ. It is more usual to represent a signal Σ of slope v by a straight line in a geometric diagram than by means of states on its space-time diagram. In a dynamical understanding of a signal we can see this slope as its speed, measured from the starting time of its periodic part.

These basic signals play a very important role in algorithmics on cellular automata, as it can be seen in [50] and [**?**]. But we may wonder whether they allow to build more complex ones. A beautiful and historical example is to be found in [50], where "Fisher's parabola" is displayed. We will here and now develop a simpler example, that already put to light some difficulties.

3.1.3 Other cellular automata constructible signals

To properly answer the above question, two attributes of cellular signals have to be introduced: their ratio and speed. The *ratio* of a signal Σ is an increasing function ρ, from $\mathbb{N}$ into $\mathbb{N}$, which gives, for each cell n, the time required by Σ to reach the cell n, for the first time, from the origin. The *speed* of Σ to reach the cell n is then $v(n) = \frac{n}{\rho(n)}$. We immediately observe that the maximal possible speed in case of von Neumann neighborhood is 1, but actually it depends, in a non trivial way, on the neighborhood.

Now let us come back to the production of the 2^n, $n \in N$, for showing how it can be transformed to give rise to a (cellular) signal of exponential ratio. Actually, that is illustrated on the geometric diagram of the Figure 9. The signal Σ' is constructed thanks to two others, D and Δ', in the following way. The three start from the origin, Σ' and D follow the same way up to $(2,4)$, there, they split: the first one goes on up, the second one goes at speed 1 up to meet, at $(3,5)$, Δ', which has followed the indicated track. At $(3,5)$ D comes back to the left at speed 1, up to meet Σ', then they run together up for one time unit, then at speed 1 for one time unit and they part as after their first meeting (the first one going up, the second one on the right at speed 1), while Δ' is going from $(3,5)$ to $(4,5)$ where it takes speed $1/3$. This process will be repeated as shown on Figure 9(a).

Actually we start from the first "natural" geometric diagram, Figure 9(a), where the basic points are the (x_n, y_n), with $x_n = n + 2^{n-1} - 1$, $y_n = 2^n + 2^{n-1} - 1$ for $n \geq 2$. But it is easy to observe that the segments between (x_n, y_n) and (x_{n+1}, y_{n+1}) don't have the same slope, while it is the case for the segments delimited by the points (x'_n, y_n) and (x'_{n+1}, y_{n+1}) where $x'_n = x_n + 1$, the slope of which is 3 (Figure 9(b)).

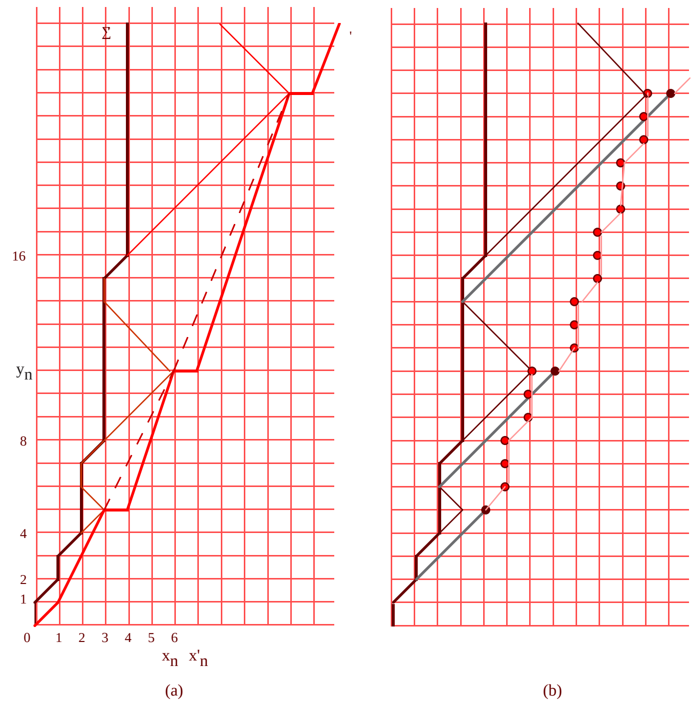


Figure 9: Geometric (a) and cellular (b) signals to obtain a cellular signal of ratio $n \mapsto 2^n$.

It remains to transform the last geometric diagram in a cellular geometric diagram, we have to find a way to mark the points (x'_n, y_n). This is done via a new signal D' as it is shown on Figure 9(b). Let us notice that D' knows that it has to die on (x'_n, y_n) one time after meeting Σ', and to be replaced from there by a signal of speed $1/3$. And finally a better solution is given on Figure 10.

But does that mean that there exists a 1-cellular automaton generating this signal of ratio $n \longmapsto 2^n$?

Remark 1.

First of all, let us emphasize that the problem to know whether for a given geometric diagram there exists a cellular automaton which realizes it is undecidable. The basic reason is that, generally, it is impossible to determine

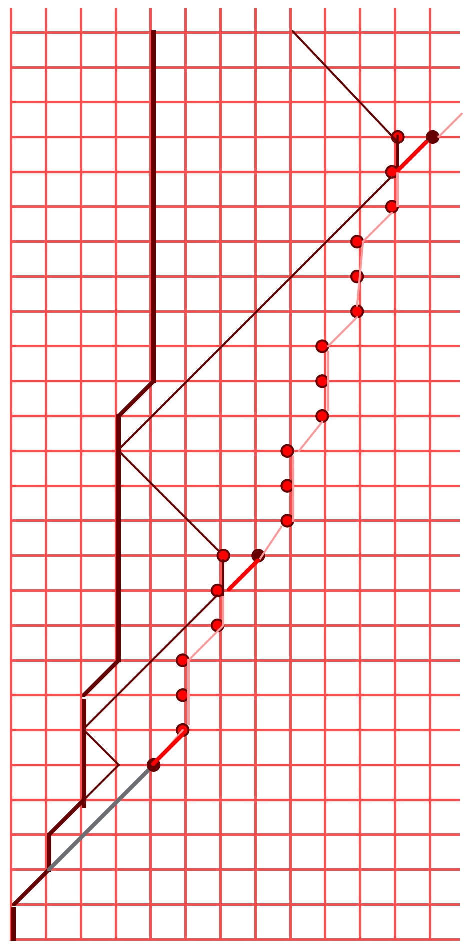


Figure 10: Cellular signal of ratio $n \mapsto 2^n$ and signals needed to build it.

if the number of different signals which can have to be superposed is finite. Actually, would this problem be decidable, would the problem to know whether a state emerges in the evolution of a cellular automaton be decidable, which is not (see [68]).

So quite each case is a singular one, and it is not always easy to design a cellular automaton (assuming that such an automaton exists) even governed under an expressive cellular geometric diagram. It is not the case for our example, and a space-time diagram of a suitable automaton is to be found on Figure 11.

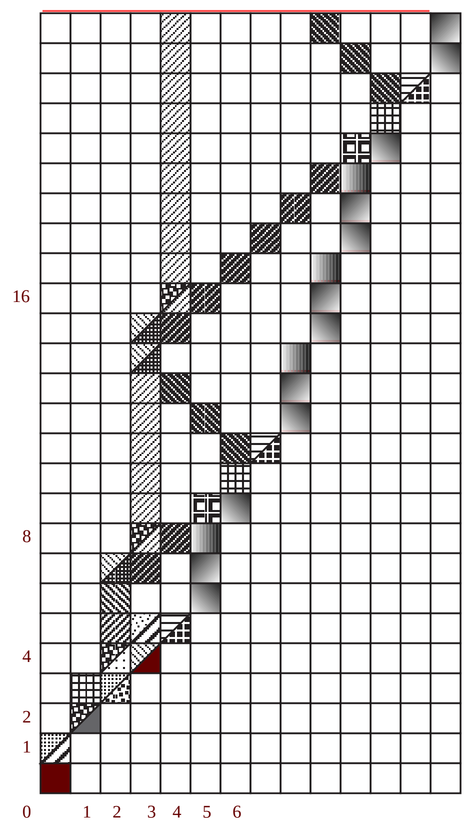


Figure 11: Part of a space-time diagram representing the construction of a signal of ratio $n \longmapsto 2^n$.

Let us mention that it is possible to build cellular signals of quadratic ratios, ratios involving roots and logarithms. The interested reader can refer to [54]. To finish this section we will still cite an interesting result of this paper, which shows that there exist gaps among the ratios of constructible signals.

Theorem 6. *A gap theorem*

For any constructible signal of ratio $n \longmapsto \rho(n)$*, either* $\rho(n) - n$ *becomes constant or the signal is lower bounded by a signal of ratio* $n + log_c(n)$ *(*c *is a constant,* $c \in \mathbb{N}$*), which means that* $n + log_c(n) \leq \rho(n)$ *for every* n*,* $n \in \mathbb{N}$*.*

3.2 Signals in dimension d, $d > 1$

Although different notions of signals commonly occur in the literature for dimension 2 (significant examples are to be found in [16] and the gliders of the "Game of Life" ([8] and [28]) can be interpreted as signals), nothing formal is known in this area, which is still to open up. It is already difficult to see inside the space-time diagram of a 2-dimensional cellular automaton. Moreover we can imagine all sorts of trajectories (curves) depending on the chosen neighborhood, or waves fronts or volumes ... In fact, when "signals" are touched on in higher dimensions, it is often via projections on the plane, which is a way of bypassing the difficulty, as in [25].

Actually the notion of signal and the use it is done of are closely connected to the problematics of time optimization, especially the problematics of *real time*, that is, roughly speaking, how to realize objectives *as soon as possible*. Then some difficulties can arise even in dimension 1.

4 Universalities

As it was already noticed computational universality and some biological-like robustness were essential von Neumann's requirements for the device he was in search of. So he became a pioneer for *computational universality* and *construction universality*, which were formalized, and deeper investigated, a little later by Burks and people around him [11], as well as by Codd [16] and Banks [5]. Their main work was to build 2-dimensional cellular automata with these properties and few states, or to find conditions for a cellular automaton to satisfy them.

Outside construction universality that will be considered later, people are, in our context, essentially or ultimately interested in computational universality. But there are several notions of computational universality, depending on the fact that they are set up by means of simulations or not, and, in the former case for example, depending upon the fact that simulations are between computation models of different nature, essentially Turing machines and cellular automata (extrinsic simulations), or between cellular automata (intrinsic simulations), or depending on the configurations considered, on the encoding or decoding understandings ...

Intrinsic universality, corresponding to some intrinsic simulations, was pointed out later in [1]. It also expresses the power of some cellular automaton with respect to other (or a class of) cellular automata, and so attests a sort of model (inner or auto-) coherence and potential expressive power.

In the sequel we present the notions of universality which are at work in the basic historical universality results. Most of them are founded on implicit simulation notions, generally "ad hoc". We put to light some related difficulties and refer to [28] for illustration and a deeper discussion.

4.1 Computational universality

What is at stake in this area is to study the computational power of the model. Finite configurations are then basic objects. Here appears with a noticeable clarity the power given to cellular automata by their infinite component (their space), but also the difficulties it arises, in particular because a computation step implies the instantaneous transformation of the infinite number of cells, so that some isolated cell can have hidden perverse effects on computation areas generated by some other very distant finite part of the initial configuration.

Roughly speaking, a cellular automaton will be computation-universal if it is able to compute any Turing-computable (or recursive) partial function. But what means "computing a function ψ" for a cellular automaton ? Still roughly, that means that there exists some initial configuration, part of which can be interpreted as an argument for ψ, that possibly evolves to an halting one, part of which can be interpreted as the ψ-image of this very argument. Three problems already arise: *how to encode the argument (in particular in such a way that the result does not appear in this encoding), how to recognize the end of a computation, how to decode the image*? The main difficulty is to suitably encode finite data into an infinite set of sites. This direct way was taken by von Neumann or more precisely by his main continuators Burks, Thatcher, Codd and Banks. Another way to tackle the problem has been to directly compare cellular automata and Turing machines computational power by means of simulations.

4.2 Direct computational universality

4.2.1 Computation universality by Codd

We will first recall the thought processes in the sixties. They end up with devices - actually special configurations of some cellular automaton -, computing the partial recursive functions. We will follow Codd [16], who set up formal definitions rendering thoughts, and their explicit achievements, at the time. These definitions are not the more general and have to be used with caution because they let subsist some difficulties (which do not affect the effective constructions). Note that Codd focused on 2D-cellular automata, but that what he established is also available for 1-dimensional ones.

Computation-universal cellular automata We now consider cellular automata with a quiescent state ν and global function G. Let us start with some technical definitions and notations.

1. If two configurations c and c' are disjoint ($Supp(c) \cap Supp(c') = \emptyset$), $(c \sqcup c')$ denotes the configuration defined by $(c \sqcup c')(x) = c(x)$ on $Supp(c)$, $(c \sqcup c')(x) = c'(x)$ on $Supp(c')$, and $(c \sqcup c')(x) = \nu$ on the complement (in the space) of $Supp(c) \cup Supp(c')$.

2. A *subconfiguration* c' of a configuration c is a configuration such that $c_{|Supp(c')} = c'_{|Supp(c')}$.

3. A configuration c is called *passive* when it is a fixpoint for G, that is $G(c) = c$, and *completely passive* when all its subconfigurations are passive.

4. A configuration c is said to pass on information to a configuration c', disjoint of c, when there exists a time t such that $G^t(c \sqcup c')_{|Supp(G^t(c'))} \neq G^t(c')_{|Supp(G^t(c'))}$.

5. A configuration c is a translation of a configuration c' if there exists a vector τ such that $c'(x) = c(x - \tau)$ for each x in the space.

To take into account the functions data, special sets of *finite configurations* are distinguished, called *Turing domains*. Let $\mathcal{A}$ be a cellular automaton, a Turing domain for $\mathcal{A}$, denoted $T_{\mathcal{A}}$, is a set of $\mathcal{A}$-configurations which satisfies the following conditions:

1. they are finite configurations and there is an effective (recursive) injective application $\iota_{\mathcal{A}}$ from N into $T_{\mathcal{A}}$,

2. each configuration in $T_{\mathcal{A}}$ is completely passive and, moreover,

3. the configurations in $T_{\mathcal{A}}$ are collectively passive, which means that for each finite subset of $T_{\mathcal{A}}$, each set of translations which yields disjoint translates, the union of these translates is completely passive.

Definition 5. *Computation-universal cellular automaton*

A cellular automaton $\mathcal{A}$ is said to be computation-universal *if there exists a Turing domain for $\mathcal{A}$ and if, for each Turing-computable partial function ψ from $\mathbb{N}$ to $\mathbb{N}$, there exists an $\mathcal{A}$-configuration c, $c \notin T_{\mathcal{A}}$, which* computes *ψ, that is that there exists a cell α, $\alpha \in Supp(c)$ and a non-quiescent state s_{halt}* [5] *such that for each n in the ψ-domain:*

1. $G^t(c \sqcup \iota_{\mathcal{A}}(n))(\alpha) = s_{halt}$,
2. $G^{t'}(c \sqcup \iota_{\mathcal{A}}(n))(\alpha) \neq s_{halt}$ *for all* $t' \leq t$
3. $G^t(c \sqcup \iota_{\mathcal{A}}(n))_{|\overline{\cup_{f \in T_{\mathcal{A}}} Supp(f)}}$ *does not pass information to* $G^t(c \sqcup \iota_{\mathcal{A}}(n))_{|\cup_{f \in T_{\mathcal{A}}} Supp(f)}$,
4. $G^t(c \sqcup \iota_{\mathcal{A}}(n))_{|\cup_{f \in T_{\mathcal{A}}} Supp(f)} = \iota_{\mathcal{A}}(\psi(n))$.

As $T_{\mathcal{A}}$ is a set of finite configurations and $\iota_{\mathcal{A}}$ is one-to-one, there is no problem of decoding. So $\mathcal{A}$ is such that each Turing-computable partial function is computed by means of one of its configurations.

Universal computers We can wonder whether it could be done by only one of them. This leads to the notion of universal computer, which is a particular configuration of some cellular automaton.

Definition 6. *Universal computer*

Let $\mathcal{A}$ be a cellular automaton with a Turing domain $T_{\mathcal{A}}$. An $\mathcal{A}$-configuration c, $c \notin T_{\mathcal{A}}$, is called a universal computer *if, for any Turing-computable partial function ψ, there exist d in $T_{\mathcal{A}}$ and a translation defined by a vector τ such that the translate d_τ of d by τ is disjoint of c, not in $T_{\mathcal{A}}$, and that $c \sqcup d_\tau$ computes ψ.*

Clearly, the existence of a universal computer for a cellular automaton makes it a computation-universal cellular automaton. But it is not known whether a computation-universal cellular automaton has necessarily a universal computer. Actually,

Fact 3.

The historical "universal" 2-dimensional cellular automata (with von Neumann neighborhood), as the von Neumann's one (29 states), the Codd's one (8 states) and the Banks' one (4 states) are cellular automata with universal computers.

[5] Another suitable way is to consider fix points as halting configurations.

Their design is not so simple (and even needs books - for the first ones - to be described and proved right). The interested reader is invited to read [16], [5] and [10].

Another way to cope with the computable partial functions was to find suitable cellular automata for each of them or, at least for the basic ones. That was recently done by J. Mazoyer.

4.2.2 Computation universality via the Kleene's recursive functions

We know that the Turing computable functions are exactly the class of partial recursive functions defined as the smallest class of partial functions that contains the constant function **0**, the successor function, the projections and is closed under substitution, recursion and minimization (see for example [22]). Very special algorithmic techniques on grids are used to establish the following result.

Fact 4. *[52]*

Each partial recursive function can be computed on a 1*-dimensional cellular automaton.*

We have no more a "universal device", but a *property of the cellular automata computation model*, that we can - a little improperly - render by 1-*dimensional cellular automata are computationally universal.*

Let us now come to notions and results founded on simulations, first between different models and, afterwards, inside the model.

4.3 Computation universality via Turing machines

It seems obvious that if every one-bi-infinite-tape Turing machine (or even better a one-bi-infinite-tape universal Turing machine) can be simulated by one cellular automaton, "cellular automata" have, at least, the Turing machines computational[6] power and so can be considered as a "computation universal model".

The main question is then what do we mean by a simulation? A lot of problems are solved by means of simulations (models comparisons, various optimization achievements, complexity issues, ...). But, although the expected results deeply depend upon them, simulations which realize, under

[6]Recall that from the computability point of view Turing machines and cellular automata - the latter ones here considered on finite configurations - are equivalent, but from the complexity point of view, cellular automata are much more efficient. See Part 3 in this volume and [81] for example.

conditions, the connection between two devices (or two classes of devices) are ad hoc or, often, underhanded. In particular, different notions of universality are attached to different sorts of simulations.

4.3.1 Simulations

A Turing machine which computes a partial function from $\mathbb{N}$ to $\mathbb{N}$ can be considered as a 4-tuple $(\mathcal{C}, \mathcal{T}, \mathcal{D}, \mathcal{E})$, where $\mathcal{C}$ denotes its configurations set, $\mathcal{T}$ its global transition function (mapping each configuration on its successor via the transition function of the machine (supposed deterministic), $\mathcal{D}$ is a function from $\{0,1\}^+$ into $\mathcal{C}$ which sends each word (coding a non negative integer) on a configuration which will be an initial configuration for the machine, and $\mathcal{E}$ a partial function from $\mathcal{C}$ to $\{0,1\}^+$ which will associate to each halting configuration a word on $\{0,1\}$ (and so, a non negative integer).

We can certainly get an analogue description for cellular automata. But a great difference with Turing machines for which configurations are finite[7] is, precisely, that for cellular automata configurations are infinite objects. Thus, finite words have to be set up on infinite spaces. This difficulty leads first to limit the set of configurations to the one of finite configurations, which requires automata with quiescent states. A computing cellular automaton can be represented by $(\mathcal{C}_f, \mathcal{G}, \mathcal{D}, \mathcal{E})$.

Now, to formalize a simulation of a Turing machine by a cellular automaton, we need a coding of the configurations of the first one into the finite configurations of the second one.

Definition 7. *Simulation of a Turing machine by a cellular automaton with a quiescent state*

Let $(\mathcal{C}_\mathcal{M}, \mathcal{T}_\mathcal{M}, \mathcal{D}_\mathcal{M}, \mathcal{E}_\mathcal{M})$ and $(\mathcal{C}_{f\mathcal{A}}, \mathcal{G}_\mathcal{A}, \mathcal{D}_\mathcal{A}, \mathcal{E}_\mathcal{A})$ be descriptions for a Turing machine $\mathcal{M}$ and a cellular automaton $\mathcal{A}$ (with a quiescent state). We say that an injective recursive application ? from $\mathcal{C}_\mathcal{M}$ into $\mathcal{C}_{f\mathcal{A}}$ is a simulation of $\mathcal{M}$ by $\mathcal{A}$, or that $\mathcal{A}$ simulates $\mathcal{M}$, if the following conditions are satisfied:

- *for each u in $\{0,1\}^+$ there exist t, t' such that $?({\mathcal{T}_\mathcal{M}}^t(\mathcal{D}_\mathcal{M}(u)))$ is ${\mathcal{G}_\mathcal{A}}^{t'}(?(\mathcal{D}_\mathcal{M}(u)))$,*
- *for each halting configuration c in $\mathcal{C}_\mathcal{M}$, there exists t such that ${\mathcal{G}_\mathcal{A}}^t(?(c))$ is a configuration c' in the $\mathcal{E}_\mathcal{A}$-domain such that $\mathcal{E}_\mathcal{A}(c') = \mathcal{E}_\mathcal{M}(c)$.*

In fact, when we are interested in the simulation complexity, we set up a more uniform definition in assuming the existence of ? and of two integers k et k' such that for each u in $\{0,1\}^+$, $?({\mathcal{T}_\mathcal{M}}^k(\mathcal{D}_\mathcal{M}(u)))$ is ${\mathcal{G}_\mathcal{A}}^{k'}(?(\mathcal{D}_\mathcal{M}(u)))$. That makes the simulation configuration-independent.

[7] Usually, or at least in the computability area.

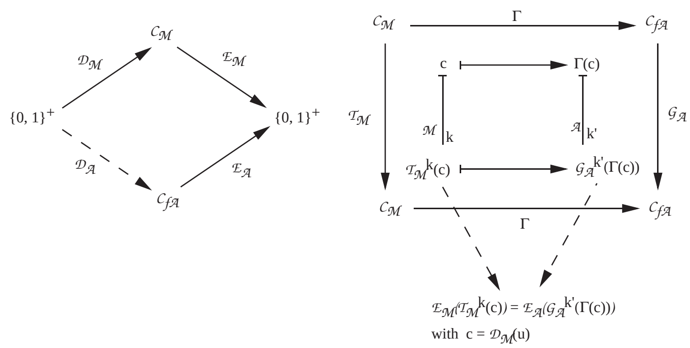


Figure 12: A simulation of Turing machines by cellular automata.

This notion is used to get results as the following ones, which are historical in this trend. In [69], Smith looks for "small" (in number of states and in number of neighbor cells) cellullar automata able to compute Turing computable functions. Let us mention here that Codd [16] got a computation-universal 2-states automaton by simulation of its (8-states, von Neumann neighborhood) automaton, but with a neighborhood of 85 cells!

Smith also proved the noticeable fact that dimension 1 is sufficient for computational universality. Let us now list some of his results which also evaluate the simulations times.

Theorem 7. *[69]*

1. *For each* (m, n) *Turing machine (m symbols, n states), there exists a* 2*-dimensional cellular automaton with* $max(m + 1, n + 1)$ *states and a neighborhood of seven neighbors, which simulates it real-time (*$k = 1 = k'$ *in the above simulation).*

2. *For each* (m, n) *Turing machine, there exists a* 2*-dimensional cellular automaton with* $max(2m + 1, 2n + 2)$ *states and the von Neumann neighborhood, which simulates it in* 3 *times real-time (*$k = 1, k' = 3$*).*

3. *For each* (m, n) *Turing machine (m symbols, n states), there exists a* 1*-dimensional cellular automaton with* $max(m + 1, n + 1)$ *states and a neighborhood of six neighbors, which simulates it real-time (*$k = 1 = k'$ *in the above simulation).*

4. *For each* (m, n) *Turing machine, there exists a* 1*-dimensional cellular automaton with* $m + 2n$ *states and the von Neumann neighborhood, which simulates it in at most* 2 *times real-time.*

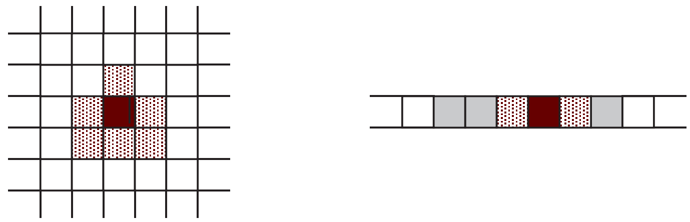


Figure 13: Seven and six cells neighborhoods of Smith.

These results, together with the existence of a $(6,6)$ universal Turing machine [56], imply the following corollaries:

Theorem 8. *[69]*

1. *There exists a* 2*-dimensional cellular automaton computation -universal with* 7 *states and a neighborhood of seven neighbors.*
2. *There exists a* 2*-dimensional cellular automaton computation -universal with* 14 *states and the von Neumann neighborhood.*
3. *There exists a* 1*-dimensional cellular automaton computation -universal with* 7 *states and a neighborhood of six neighbors.*
4. *There exists a* 1*-dimensional cellular automaton computation -universal with* 18 *states and the von Neumann neighborhood.*

A lot of results have been obtained since, in the same vein, aiming to evaluate and optimize the simulations in number of states (the search for small universal cellular automata is still alive), or to restrict the class in which candidates to be universal lay (see 4.4.4.).

Other proofs of the universal computational power of cellular automata can be found. Let us cite, for example, that *every two-pushdown automaton can be simulated by a* 1*-dimensional cellular automaton* [35], which allows the wanted conclusion because a two-pushdown automaton is equivalent to (has the computational power of) a Turing machine.

Note that in dimension 2, the proof of computation-universality of the "Game of Life" by Conway is in a similar vein.

Carrying on with work and reflections initiated by Wolfram [85], some people try to find out connections between computational universality and complexity [81].

The above results lean on simulations of sequential models expressing computational universality, which masks or neglects the parallel nature of cellular automata. Moreover it is interesting to compare the expressive power of cellular automata on their own. This leads to *intrinsic universality.*

4.4 Intrinsic universality

The basic question is: do there exist cellular automata able to simulate all cellular automata, at least of a given class? It is not a problem to canonically code the finite automaton underlying each cellular automaton, and to get a recursive enumeration of them. The problem arises with the configurations, which are infinite ... So, once more, if we restrict the devices to their finite configurations, we can imagine a solution analogous to the solution for Turing machines: find a cellular automaton $\mathcal{U}$ such that, given the code of any automaton $\mathcal{A}$ of a given class and the code of a finite configuration c of $\mathcal{A}$, we can get, in a configuration of $\mathcal{U}$, the halting configuration of $\mathcal{A}$ on c when it exists (a serious problem will then be to satisfactorily formulate the notion of halting configuration).

4.4.1 Intrinsic universality in the style of Čulik

As it should now be understood, universality is not an intrinsic notion! Most of time it is, at least, indissociable from a simulation notion and a class of automata. So we will first give the definition of a suitable simulation.

Definition 8. *I_1-Simulation*

Let $\mathcal{CA}$ be a class of cellular automata and $\mathcal{CA}_f$ the set of their finite configurations. We will call I_1-simulation (for intrinsic simulation in sense 1) any triple (Φ, χ, π) of recursive functions, Φ from $\mathcal{CA}$ to itself, χ from $\mathcal{CA}\times\mathcal{C}_{\mathcal{A}f}$ into $\mathcal{C}_{\mathcal{A}f}$, π from $\mathbb{N}$ to $\mathbb{N}$, such that for each automaton A, each finite configuration c for A, if c_t is the configuration obtained in A after t computation steps starting with c, then $\chi(A, c_t)$ is the configuration obtained in $\Phi(A)$ after $\pi(t)$ computation steps starting with $\chi(A, c)$, that is $G^{\pi(t)}_{\Phi(A)}(\chi(A, c)) = \chi(A, G^t_A(c))$.

To this sort of simulation is associated a notion of universality, the I_1-universality.

Definition 9. *I_1-universal cellular automaton*

A cellular automaton $\mathcal{U}$ is said to be I_1-universal for the class $\mathcal{CA}$ if there is an I_1-simulation (Φ, χ, π) such that, for each A in $\mathcal{CA}$, $\Phi(A) = \mathcal{U}$.

A first result in this area was the following one [1], which asserts the existence of a I_1-universal cellular automaton for the class of the 1-dimensional-one-way cellular automata.

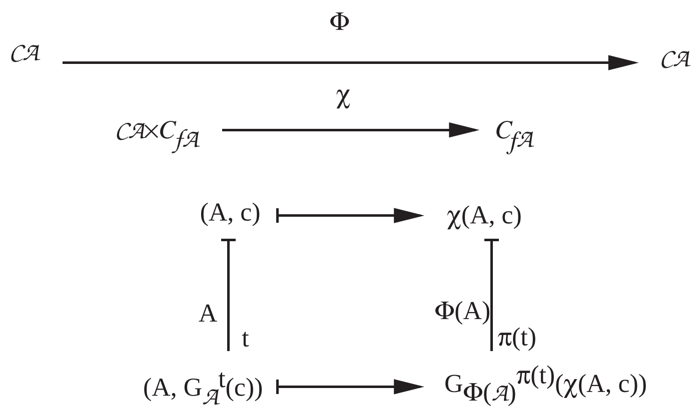


Figure 14: A first simulation: I_1-Simulation between cellular automata.

Theorem 9. *Albert-Čulik theorem*

There is a 1*-dimensional cellular automaton (*14 *states, first neighbors neighborhood), which I_1-simulates any totalistic* 1*-dimensional-one-way cellular automaton.*

This result was extended to any 1-dimensional cellular automaton in [49]. And it is worthy to notice that a consequence of this result is that, by composing simulations, computation-universality only needs one dimensional totalistic automata.

But, nevertheless, this last notion of intrinsic universality is not really satisfying because of the restriction to the finite configurations (but if we are interested in computing, they are difficult to by pass) and because the components of the simulation are not homogeneous to the model: they can not be interpreted as global functions of cellular automata. So we are going to set up another more suitable definition of simulation, more like the one given for Turing machines.

4.4.2 Intrinsic universality in an other style

This naturally starts with the definition of a new notion of simulation.

Definition 10. *I_2-simulation*

Let $\mathcal{CA}$ a class of cellular automata, $\mathcal{C}$ their configurations set. We say that a cellular automaton $\mathcal{B}$ in $\mathcal{CA}$ I_2-simulates $\mathcal{A}$ of $\mathcal{CA}$ if there exists an injective application ? from $\mathcal{C}$ into itself such that for each configuration c of $\mathcal{A}$ there exist t, t' such that $?(G^t_{\mathcal{A}}(c)) = G^{t'}_{\mathcal{B}}(?(c))$.

In fact we consider uniform simulations in which there exist ? and integers k and k' such that $?(G^k_{\mathcal{A}}(c)) = G^{k'}_{\mathcal{B}}(?(c))$, and, in particular, the case where

$k = 1$ ([67]).

$$\begin{tikzcd}
\mathcal{CA} \arrow[rrr, "\Gamma"] & & & \mathcal{CB} \\
& c \arrow[r, mapsto] \arrow[d, mapsto, "\mathcal{A}"', "k"] & \Gamma(c) \arrow[d, mapsto, "\mathcal{B}"', "k'"] & \\
& G_{\mathcal{A}}^{k}(c) \arrow[r, mapsto] & G_{\mathcal{B}}^{k'}(\Gamma(c)) &
\end{tikzcd}$$

Figure 15: Another simulation: I_2-Simulation between cellular automata.

To this simulation is also attached a notion of universal cellular automaton.

Definition 11. *I_2-universal cellular automaton*
A cellular automaton of the class $\mathcal{CA}$ is I_2-universal for this class if it I_2-simulates each cellular in $\mathcal{CA}$.

Without any further hypothesis on ? or on the configurations, we take the risk not to find any I_2-universal cellular automaton. So we have to be a little more precise. If we restrict the configurations to finite ones and if we demand ? to be recursive, then we have examples. But better can be done, and for a more complete discussion we refer the reader to [28] where the Game of Life is there proved to be, in fact, intrinsically universal.

We will just put to light that if we ask in the above definition that $k = 1, k' = T$ with some T independent of the configuration and that ? were the global function G of some cellular automaton of $\mathcal{CA}$, the condition of the definition would be $G(G_{\mathcal{A}}(c)) = G_{\mathcal{B}}^{T}(G(c))$, then $G \circ G_{\mathcal{A}} = G_{\mathcal{B}}^{T} \circ G$. Such a commutation property is connected with the notion of grouped cellular automata we will present in the next section.

4.4.3 Connections between the above notions of universality

We have brought to the fore some (main) notions of universality, without looking for exhaustiveness, but to recall that universality for cellular automata is multifaceted, designates properties which, most of time, are not equivalent and whose connections, when they exist, are often founded on delicate codings and implicit restrictions on hypotheses. Up to now most of results refer to computational universality in their different senses. But the intrinsic universality issue seems to be noteworthy. We have also wanted to draw attention to the fact that sometimes universality characterizes special configurations, cellular automata, classes of cellular automata or even classes of automata on particular configurations.

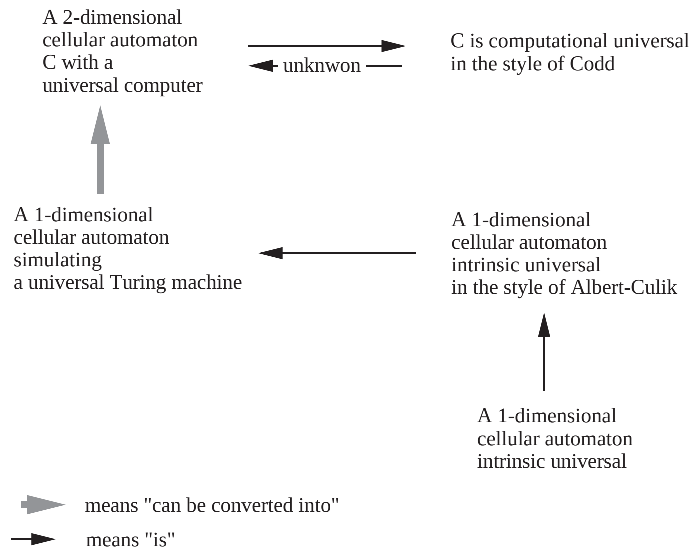


Figure 16: Connections between different notions of universality.

4.4.4 Minimal computation universal cellular automata

To finish with universality, let us notice that the search of universal automata with minimal number of states still goes on. But, often, less states seems to imply some compensation for example a bigger neighborhood (dimension 2, 2 states but a neighborhood with 85 cells in [16]), or some background (dimension 2, 2 states, von Neumann neighborhood and a background [5]). See also [46] which recall or prove significant results, and [30] which is recent in the race. A *background* for a 1-dimensional cellular automaton is a configuration c which is both time and space periodic. Time-periodic means that $(G^t(c))_{t\geq 0}$ is periodic, and, if σ denotes the shift on configurations (defined by $(\sigma(x))_i = x_{i+1}$), c is space periodic when the sequence $(\sigma^n(c))_{n\geq 0}$ is periodic.

Another notion of universality, rather historical than widely or even rigorously developed, but historically interesting, is construction universality, which is often (and often wrongly) mixed up with (self-)reproduction.

4.5 Construction universality

Once more, let us come back to von Neumann who was interested in understanding the logical organization of natural complex systems, especially systems able of auto-reproduction. He wondered what is necessary for an "automaton" to reproduce itself, without losing any of its properties, especially its self-reproducing ability. This concern, together with the one of computational power he required, leads him to conceive its automaton, and to set up this idea of construction universality.

4.5.1 Construction universality in the von Neumann's style

Still following Codd we will say that a configuration c of an automaton $\mathcal{A}$ builds some $\mathcal{A}$-configuration c' if there exists some time t such that c' is a subconfiguration of $G^t_{\mathcal{A}}(c)$, disjoint of c and such that $G^t_{\mathcal{A}}(c) - c'$ does not pass on information to c'. This definition set aside fix points, which would be trivial, and assures its autonomy to the constructed configuration. But these conditions are not sufficient to avoid the risk of triviality (think to the shift or to $(\{0,1\},\delta)$ with $\delta(x,y,z) = x+y+z \pmod 2$ on the initial configuration $0^\omega 10^\omega$). It is why computation abilities were also required.

Let $\mathcal{U}$ be a computation-universal cellular automaton (in the Codd sense) and suppose there exists a set C of configurations such that each of them computes a Turing-computable partial function and such that, conversely, each Turing-computable partial function were computed by a member of C. If every element of C is constructible by some configuration disjoint from C, then $\mathcal{A}$ is said to be *construction-universal*. When one fixed configuration constructs each member of C, it is called a *universal constructor*, and is certainly non trivial since it constructs significant configurations. It is really powerful when it is also a universal computer. But let us notice that a constructor can be a universal constructor without being able to build itself.

Fact 5.
The 2-dimensional von Neumann's and Codd's cellular automata are construction universal, with universal computer-constructors, which happen to also be self-reproducing.

Before coming to self-reproduction, let us remark that this notion of construction universality has been interpreted in a different way, in fact including a condition of self-reproduction.

4.5.2 Construction universality in other styles

Let us say that a configuration c' is a copy of a configuration c if there exists a translation τ such that $c' = c_\tau$.

Some authors consider that a cellular automaton $\mathcal{A}$ is construction universal when there exists a finite configuration u of $\mathcal{A}$ satisfying the following conditions:

- there is a time t such that $G^t_{\mathcal{A}}(u)$ contains two disjoint copies of u,
- for each finite configuration c of $\mathcal{A}$, disjoint of u, there is a time t such that $G^t_{\mathcal{A}}(u \sqcup c)$ contains a copy of c, disjoint of c.

Under this definition, we can assert that a cellular automaton with Garden-of-Eden configurations can not be construction-universal, and, in that frame, the "Game of Life" is an example of a cellular automaton computation-universal and self-reproducing but not construction-universal.

Actually, the idea which has rather held researchers attention was, and still is, the self-reproduction one, although what is understood under this name often is woolly enough.

4.6 Self-reproduction

Von Neumann was mainly interested in the ability of complex natural, living systems, to withstand some damages (in repairing them or not) and, so, to keep their capacities. This certainly motivated his investigations on construction and reproduction, and the conception of his device.

Afterwards many works have been dedicated to "self-reproduction". It is question of configurations producing copies of themselves or replications keeping or not their ability of (re)production. The attraction for the terminology is certainly due to its biological connotation, although it, often, only covers some replication ability.

4.6.1 Self-reproduction in the style of Moore

A first paper by Moore [59] in this domain points out some difficulties about the notion (which can be trivial: let us consider the configuration with 1 on cell 0 and 0 everywhere else under the right shift), shows the limitations induced by the finiteness of the representation and by the underlying space itself.

Definition 12. *Self-reproducing configuration*

- *A configuration c' of some cellular automaton $\mathcal{A}$ is said to contain n copies of some configuration c if c' has n, $n \in N$, disjoint subconfigurations, each of which being a copy of c.*
- *An initial configuration c is said to be capable of reproducing n offspring by time T, if there is a time T', $T' < T$, such that ${G_{\mathcal{A}}}^{T'}(c)$ contains at least n copies of c*

- *A configuration c is said to self-reproduce or to be self-reproducing if, for each positive integer n there is a time T such that c is capable of reproducing n offspring by time T.*

We have to keep in mind that this definition does not put aside the risk of triviality. Moreover the following Moore's result shows that the population generated by this former process of self-reproduction does not satisfy the, generally assumed, exponential growth's law of living organisms.

Theorem 10.
If a self-reproducing configuration is capable of reproducing $f(T)$ offspring by time T, then there exists a positive real number k such that $f(T) \leq kT^2$.

The paper by Moore ends up in putting to light the Garden-of-Eden configurations and proving a sufficient condition for Garden-of-Eden configurations existence via erasing configurations.

This definition is still used by Smith, who gives in [70] a non constructive, but short, proof of the existence of non-trivial self-reproducing configurations. The proof rests on the existence of universal Turing machines, the classical Kleene's recursion theorem [66] and the technique of "wiring-in" Turing machines into cellular automata already used in [69].

Theorem 11.
There exists a 1-dimensional cellular automaton $\mathcal{A}$ and a $\mathcal{A}$-configuration c such that c self-reproduces and is computation-universal.

Another interest of the theorem proof, noted by the author himself, gives some light on the notion of self-reproduction at work: "*the proof reduces the problem of self-construction to a computation problem, which means that no machinery beyond ordinary computation theory is required by self-reproduction*".

But, if computation-universality is a sufficient condition for self-reproduction, it is not a necessary one, as it is shown by the well-known example of Langton [45]. It is a 2-dimensional cellular automaton, with few states, a simple non-trivial configuration c_0 of which (a loop of $15x - cells$, $10y - cells$, for tens of thousands for von Neumann or Codd ...) is capable of constructing itself periodically, or which appears periodically in the space as subconfiguration. The number of copies increases with time, indefinitely, moving further and further away, on the border (and in a given space part of it) of a special regular pattern, which limits the future evolution of the c-copies. In particular, the full configuration context of such a copy prevents it to behave as its very ancestor, who generates a very rigid population ... Can it then be spoken of "self-reproduction"?

Nevertheless, what is interesting here is that two principles are clearly at work in the evolution of c_0: an external one, the local cellular automaton transition function (or "universe law") and an inner one, the structure of $Supp(c_0)$ which allows it to construct itself in this universe.

4.6.2 Self-reproduction continued

The development of Artificial Life has reactivated researches about self-reproduction, as can be seen in the proceedings of the Santa Fe Institute Studies in the Sciences of Complexity. Let us also point out some works by and around K. Morita, who, for example, designed in [58] a cellular automaton with only 12 states, in which different shapes (worms or loops) can self-reproduce.

5 Groupings

The idea of changing the time-scale in considering grouped cells appears in [29] to set up and prove the Fischer's algorithm for prime numbers. Here, we will first present the notion of *grouped cellular automaton*, in case of dimension 2 and Moore's neighborhood which is significant, of easy access and can also easily be transposed in any dimension and any reasonable neighborhood. Afterwards we will show how this notion allows to found an algebraic classification of cellular automata. Finally, we will present a notion of grouping cellular automaton which leads mechanically to grouped automata.

5.1 Grouped Automata

Let $\mathcal{A}$ be a 2-dimensional cellular automaton, E a connected subset of a given $\mathcal{A}$-configuration. If the states of E are known, it is possible to know the states of a subset of E, after one or several units of time. For example, if E is a $m \times m$-square, $m \geq 3$, at the next time, the states of a $(m-2)\times(m-2)$-square inside the previous one are known, see figure 17. This leads to functions, that in turn allow to define the grouped automaton stemming from $\mathcal{A}$.

Definition 13.

We denote $\tilde{\delta}_m$ the application which associates to m^2 states of $\mathcal{A}$, $(m-2)^2$ states of $\mathcal{A}$, defined as follows:

- *if $m = 3$, $\tilde{\delta}_m = \delta$,*
- *if $m > 3$ and, if we denote the m^2 states*

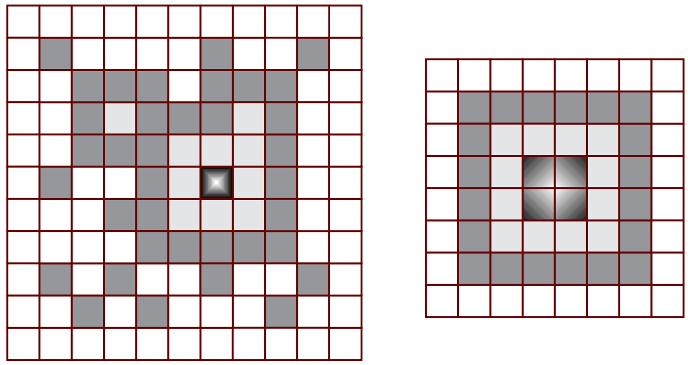


Figure 17: Marked cells at time t, $t+1$ and $t+2$.

$$\begin{pmatrix} q_{1,1} & \dots & q_{1,m} \\ \vdots & & \vdots \\ q_{m,1} & \cdots & q_{m,m} \end{pmatrix},$$

then their $\tilde{\delta}_m$-image is

$$\begin{pmatrix} \delta\begin{pmatrix} q_{1,1} & q_{1,2} & q_{1,3} \\ \vdots & & \vdots \\ q_{3,1} & q_{3,2} & q_{3,3} \end{pmatrix} & \dots & \delta\begin{pmatrix} q_{1,m-2} & q_{1,m-1} & q_{1,m} \\ \vdots & & \vdots \\ q_{3,1} & q_{3,2} & q_{3,3} \end{pmatrix} \\ \vdots & & \vdots \\ \delta\begin{pmatrix} q_{m-2,1} & q_{m-1,2} & q_{m,3} \\ \vdots & & \vdots \\ q_{m,1} & q_{m,2} & q_{m,3} \end{pmatrix} & \dots & \delta\begin{pmatrix} q_{m-2,m-2} & q_{m-2,m-1} & q_{m-2,m} \\ \vdots & & \vdots \\ q_{m,m-2} & q_{m,m-1} & q_{m,m} \end{pmatrix} \end{pmatrix}.$$

Let us observe that if we know nine $n \times n$ squares, in a configuration as the one of a cell with its Moore's neighborhood, we know the states of the central square after n successive applications of convenient functions $\tilde{\delta_m}$ with $m \leq 3n$. That determines a new function $\delta^{n,n}$ from $\left(S^{n^2}\right)^9$ into S^{n^2}, which is the composition of n $\tilde{\delta_m}$-type functions. Actually, $\delta^{n,n}$ is $\tilde{\delta}_{n+2} \circ \tilde{\delta}_{n+4} \circ \ldots \circ \tilde{\delta}_{3n}$.

Definition 14. *n-grouped automaton*
Let $\mathcal{A} = (S, \delta)$ be a 2-dimensional cellular automaton. The 2-dimensional cellular automaton $\left(S^{n^2}, \delta^{n,n}\right)$ is called the n-grouped automaton of $\mathcal{A}$, and will be denoted $G_n(\mathcal{A})$.

This definition is purely syntactic. Writing the states of $G_n(\mathcal{A})$ as matrices is a convenient way to describe them, that allows to naturally simulate the evolutions of $\mathcal{A}$ and $G_n(\mathcal{A})$. From a configuration c of $\mathcal{A}$ and an origin point (k,l) cleverly chosen, we get a tiling of c in $n \times n$ squares, each of them representing a $G_n(\mathcal{A})$-cell in a given state, then we get a configuration of $G_n(\mathcal{A})$. Actually, k and l being two given integers, there exists a bijection, we will denote $\Phi_{k,l,n}$, from the set of the $\mathcal{A}$-configurations set onto the set of the $G_n(\mathcal{A})$-ones, defined by:

$$\Phi_{k,l,n}((q_{(z,z')})_{z,z' \in \mathcal{Z}} =$$
$$\left(\begin{pmatrix} q_{nz-k,nz'-l} & \dots & q_{nz-k,nz'-l+(n-1)} \\ \vdots & & \vdots \\ q_{nz-k+(n-1),nz'-l} & \dots & q_{nz-k+(n-1),nz'-l+(n-1)} \end{pmatrix}\right)_{z,z' \in \mathcal{Z}}.$$

It holds:

Proposition 1.

Let k, l, n, be given integers, $n > 0$. Then, for each configuration c of $\mathcal{A}$, each integer t, $t \geq 0$,

$$\Phi_{k,l,n}(c^{nt}) = (\Phi_{k,l,n}(c))^t.$$

There are many ways to interpret this last result. Actually, it means that $G_n(\mathcal{A})$ simulates $\mathcal{A}$ with an acceleration of factor n. Moreover space interpretations (homotheties) are also possible as it can be seen in [25], where the method is applied to transform some bundle of discrete parabolas into an other one. We will come back later on an other grouping method, set up in order to improve some simulations results. But, it seems that the notion of grouped automaton is interesting for itself because it could give a good means of comparing cellular automata via their grouped instances. So, we will now indicate how this grouping idea give birth to an algebraic way of classifying one-dimensional cellular automata.

5.2 An order on cellular automata

Looking at space-time diagrams of 1-dimensional cellular automata from the point of view of complexity brings to try to grasp some "dominant features" and then to "forget some details " (or what can appear some time as such). That leads for example to pay attention to groups of sites rather than sites themselves and so, sometimes, to put to light macroscopic significant behaviors. See Figure 18.

We can easily translate the above subsection definitions to dimension 1. The functions $\tilde{\delta}_m$ are now defined from S^m into S^{m-2}, and if we consider blocks of n cells, the n-grouped automaton $G_n(\mathcal{A})$ of $\mathcal{A} = (S, \delta)$ is still

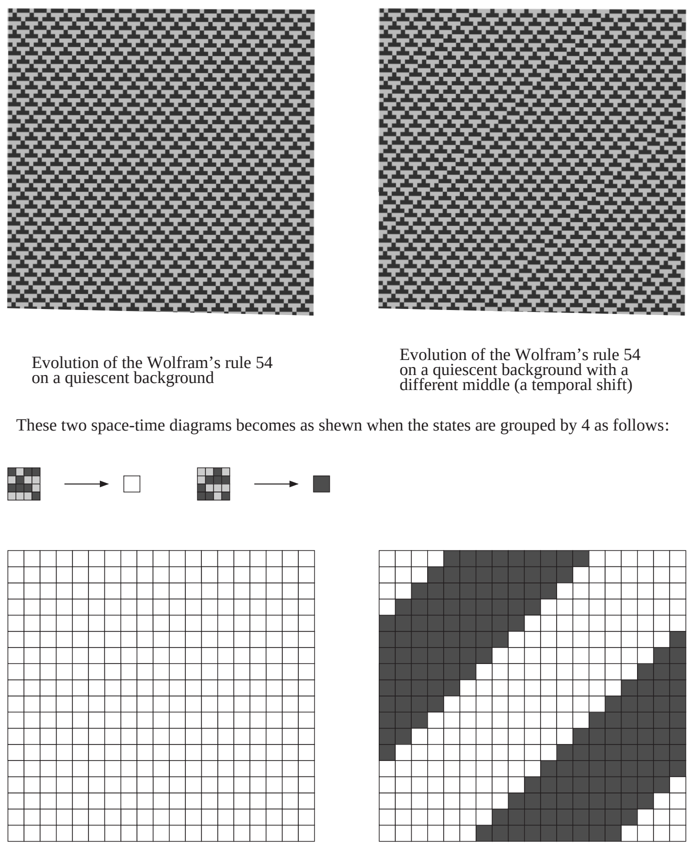


Figure 18: Grouping for the Wolfram's 54 rule.

(S^n, δ^n) with $\delta^n = \tilde{\tilde{\delta}}_{n+2} \circ \tilde{\tilde{\delta}}_{n+4} \circ \ldots \circ \tilde{\tilde{\delta}}_{3n}$. As we obtain the n-grouped automaton defined in [53], from now on, $G_n(\mathcal{A})$ will be denoted by $(S, \delta)^n$. In the latter paper a binary relation is set up on the class of 1-dimensional cellular automata by $(S_1, \delta_1) \preceq (S_2, \delta_2)$ if and only if there are non negative integers i and j such that $(S_1, \delta_1)^i \subseteq (S_2, \delta_2)^j$, which means that there exists an injection ϕ from S_1 into S_2 that satisfies $\delta_2(\phi(q_1), \phi(q_2), \phi(q_3)) = \phi(\delta_1(q_1, q_2, q_3))$.

The relation $\preceq$ is a preorder and, if $\sim$ is the canonically associated equivalence relation, $\preceq$ induces an order on the equivalence classes, which is in the process of being studied, and already proved undecidable. Some results summarized below are represented on Figure 19.

- There is a minimum element, but no maximum.
- On the first level of infinite width, some classes as NIL, PER, R_{shift}, L_{shift} capture some interesting properties.
- The existence of two infinite bounded chains, with an upper bound, which depends on a solution of some firing squad problem.
- The existence of two infinite incomparable chains.

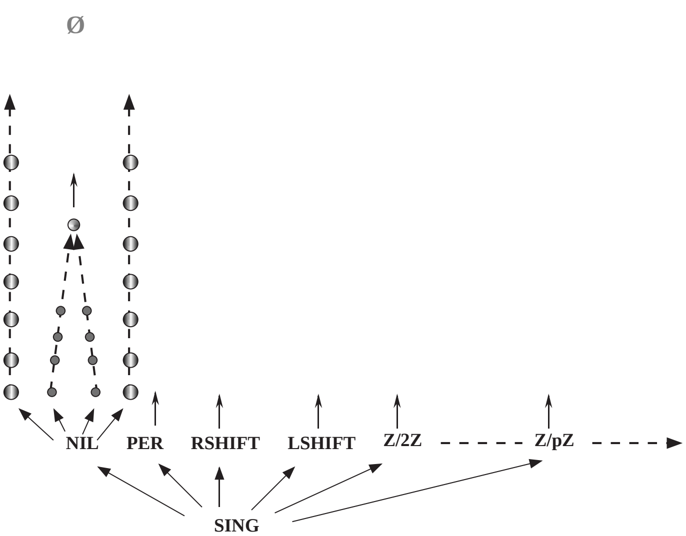


Figure 19: Some properties of the order induced by $\preceq$ on the classes modulo $\sim$.

We will now consider an other notion of grouping.

5.3 Grouping cellular automata

When observing the space-time diagrams of some language recognizers for example, the idea arises that the decision could be known earlier in suitably "twisting" them, or at least part of them. The concern for getting the possible transformations mechanically and inside the cellular automata field ends up in the notion of grouping cellular automaton [38].

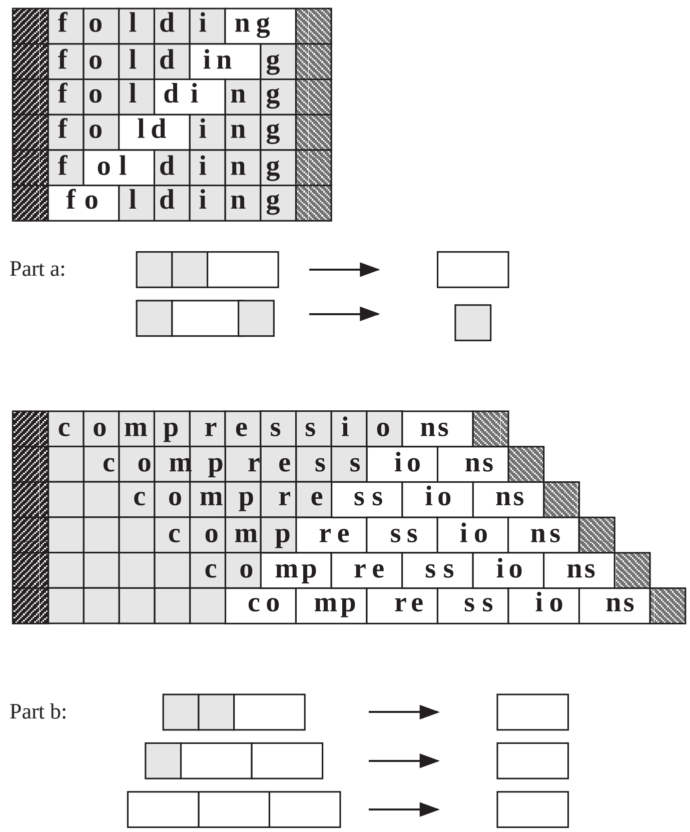


Figure 20: Some examples of folding areas which accelerate the recognition of words.

So, examining the space-time diagrams of some one-dimensional cellular automaton $\mathcal{A}$ put to light some efficient states groupings, the layout of which draws geometric pieces of the cellular space, that are in turn taken as states for the "grouping cellular automaton" $\mathcal{P}_{\mathcal{A}}$, and finally $\mathcal{P}_{\mathcal{A}}$ is made acting on $\mathcal{A}$, its pieces being then "filled" with $\mathcal{A}$-states in such a way that the obtained grouped cellular automaton, $\mathcal{A}\otimes_{\mathcal{A}}\mathcal{P}_{\mathcal{A}}$, gives the expected result.

But, unhappily, that sort of grouping does not allow to define some intrinsic simulation.

6 A last remark on terminology

Cellular automata are given various names through the literature according to the way they are used, mainly as computation models or models of natural phenomena, but also in well defined research areas. Let us cite tessalations structures, iterative circuits [11], pattern-manipulation systems [69], iterative arrays [17] ...

Moreover, another type of mixture can arise. So, as *machine* can be understood a cell, the finite automaton (S, δ), the set of "active" cells or the global machinery $(\mathbb{Z}^d, S, N, \delta)$, as well as some configuration. And the *cellular space* $(\mathbb{Z}^d)$ can be interpreted as a *universe*, a *world*, while the δ function designates the *laws* of this universe and the corresponding configurations *automata living in it.*

References

[1] Albert J. and Čulik II K. A simple universal cellular automaton and its one-way totalistic version. *Complex Systems.* **Vol. no. 1**: 1-16, 1987.

[2] Allouche J.-P., v. Haeseler F., Peitgen H.-O. and Skordev G. Linear cellular automata, finite automata and Pascal's triangle. *Discrete Appl. Math.* **Vol. no. 66**: 1–22, 1996.

[3] Allouche J.-P., v. Haeseler F., Peitgen H.-O., Petersen A. and Skordev G. Automaticity of double sequences generated by one-dimensional cellular automata. To appear in *Theoretical Computer Science.*

[4] Aso H. and Honda N. Dynamical Characteristics of Linear Cellular Automata. *Journal of Computer and System Sciences.* **Vol. no. 30**: 291–317: 1985.

[5] Banks E. R. Universality in Cellular Automata. *I.E.E.E. Ann. Symp. Switching and Automata Theory.* Santa Monica, **Vol. no. 11**: 194–215, 1970.

[6] Bartlett R. and Garzon M. Monomial Cellular Automata. *Complex Systems.* **Vol. no. 7**: 367–388, 1993.

[7] Bartlett R. and Garzon M. Bilinear Cellular Automata. *Complex Systems.* **Vol. no. 9**: 455–476, 1995.

[8] Berlekamp E., Conway V., Elwyn R. and Guy R. *Winning way for your mathematical plays, volume 2*, Academic Press, 1982.

[9] Blanchard F., Kůrka P. and Maass A. Topological and measure-theoretic properties of one-dimensional cellular automata. *Physica D.* **Vol. no. 103**: 86–99, 1997.

[10] Burks E. *Essays on Cellular Automata*, University of Illinois Press, 1972.

[11] Burks E. *Theory of Self-reproduction*, University of Illinois Press, Chicago, 1966.

[12] Byl J. Self-reproduction in small cellular automata. *Physica D.* **Vol. no. 34**: 295–299, 1989.

[13] Cattaneo G., Formenti E. and Margara L. Topological definitions of deterministic chaos. in *Cellular Automata: a parallel model*, Delorme M. and Mazoyer J. Eds, Kluwer: 213–259. To appear.

[14] Cayley A. Theory of groups. *American Journal of Mathematics.* **Vol. no. 1**: 50–52, 1878.

[15] Choffrut C. and Čulik II K. On real-time cellular automata and trellis automata. *Acta Informatica.* **Vol. no. 21**: 393–407, 1984.

[16] Codd E.F. *Cellular Automata*, Academic Press, New York, 1968.

[17] Cole S. Real-time computation by n-dimensional iterative arrays of finite-state machine. *IEEE Trans. Comput.***Vol. no. C-18**: 349–365, 1969.

[18] Čulik II K. Decisions problems on global cellular automata. in *Cellular Automata: a parallel model*, Delorme M. and Mazoyer J. Eds, Kluwer: 331–344. To appear.

[19] Čulik II K. and Yu S. Undecidability of CA classification schemes. *Complex Systems.* **Vol. no. 2**: 177–190, 1988.

[20] Čulik II K., Pachl J. and Yu S. On the limit sets of cellular automata. *SIAM J. Comput.* **Vol. 18 no. 4**: 831–842, 1989.

[21] Čulik II K., Dube S. Fractal and Recurrent Behavior of Cellular Automata. *Complex Systems.* **Vol. no. 3**: 253–267, 1989.

[22] Cutland N. *Computability*, Cambridge University Press, 1980.

[23] Delorme M. and Mazoyer J. Reconnaissance de Langages sur Automates Cellulaires. *Research report LIP-IMAG*,**Vol. no. 94-46**, 1994.

[24] Delorme M. and Mazoyer J. An overview on language recognition on one dimensional cellular automata. in *Semigroups, Automata and Langages.* World Scientific, J.Almeida Ed: 85-100, 1996.

[25] Delorme M., Mazoyer J. and Tougne L. Discrete parabolas and circles on 2D-cellular automata. To appear in *Theoretical Computer Science.*

[26] Dubacq J.-C. How to simulate Turing machines by invertible one-dimensional cellular automata. *International Journal of Foundations of Computer Science*, **Vol.6 no. 4**: 395–402, 1995.

[27] Durand B. Global properties of 2D-cellular automata. in *Cellular Automata and Complex Systems*, Golès E. and Martinez S. Eds, Kluwer, 1998.

[28] Durand B. and Róka Zs. The Game of Life: universality revisited. in *Cellular Automata: a parallel model*, Delorme M. and Mazoyer J. Eds, Kluwer: 51–74. To appear.

[29] Fischer P.C. Generation of primes by a one dimensional real time iterative array. *Journal of A.C.M.* **Vol. no. 12**: 388–394, 1965.

[30] Gajardo A. Universality in a 2-dimensional cellular space with a neighborhood of cardinality 3. *Preprint*, 1998.

[31] Garzon M. Cyclic Automata. *Theoretical Computer Science*, **Vol. no. 53**: 307–317, 1987.

[32] Garzon M. Cayley Automata. *Theoretical Computer Science*, **Vol. no. 103**: 83–102, 1993.

[33] Garzon M. *Models of Massive Parallelism*, Springer, 1995.

[34] Godsil C. and Imrich W. Embedding graphs in Cayley graphs. *Graphs and Combinatorics*, **Vol. no. 3**: 39–43, 1987.

[35] Goles E., Maass A. and Martinez S. On the Limit Set of some Universal Cellular Automata. *Theoretical Computer Science*, **Vol. no. 110**: 53–78, 1993.

[36] Head T. One-dimensional cellular automata: injectivity from unambiguity. *Complex systems*, **Vol. no. 3**: 343–348, 1989.

[37] Hedlund G. Transformations commuting with the shift. In: *Topological Dynamics*, eds. Auslander J. and Gottschalk W. G. (Benjamin, New York, 1968), 259; Endomorphisms and automorphisms of the shift dynamical system. *Math. Syst. Theor.* **Vol. 3 no. 320**, 1969.

[38] Heen O. Economie de ressources sur Automates Cellulaires. *Diplôme de Doctorat*, Université Paris 7 (in french), 1996.

[39] Hurd L. Formal language characterizations of cellular automata limit sets. *Complex Systems* **Vol. 1 no. 1**: 69–80, 1987.

[40] Ibarra O., Jiang T. and Vergis A. On the power of one-way communication. *J.A.C.M* **Vol.35 no. 3**: 697-726, 1988.

[41] Ibarra O. and Jiang T. On one-way cellular arrays. *S.I.A.M J. Comput.* **Vol.16 no. 6**: 1135-1154 ,1987.

[42] Ishii S. Measure theoretic approach to the classification of cellular automata. *Discrete Applied Mathematics* **Vol. no. 39**: 125–136, 1992.

[43] Jen E. Linear cellular automata and recurrence systems in finite fields. *Comm. Math. Physics* **Vol. no. 119**: 13–28, 1988.

[44] Kari J. Reversibility and surjectivity problems of cellular automata. *Journal of Computer and System Sciences*, **Vol. no. 48**: 149–182, 1994.

[45] Langton C. Self-Reproduction in Cellular Automata. *Physica*, **Vol. no. 100**: 135–144, 1984.

[46] Lindgren K. and Nordahl M. Universal computation in simple one dimensional cellular automata. *Complex Systems* **Vol. no. 4**: 299–318, 1990.

[47] Machí A. and Mignosi F. Garden of Eden configurations for cellular automata on Cayley graphs of groups. *SIAM Journal on Discrete Mathematics*, **Vol. no. 6**: 44–56, 1993.

[48] Martin O., Odlyzko A. and Wolfram S. Algebraic properties of cellular automata. *Comm. Math. Phys.*, **Vol. no. 93**: 219–258, 1984.

[49] Martin B. A universal automaton in quasi-linear time with its s-n-m form. *Theoretical Computer Science* **Vol. no. 124**: 199–237, 1994.

[50] Mazoyer J. Computations on cellular automata. in *Cellular Automata: a parallel model*, Delorme M. and Mazoyer J. Eds, Kluwer: 303–319. To appear.

[51] Mazoyer J. Computations on grids. in *Cellular Automata: a parallel model*, Delorme M. and Mazoyer J. Eds, Kluwer: 303–319. To appear.

[52] Mazoyer J. Computations on one-dimensional cellular automaton. *Annals of Mathematics and Artificial Intelligence* **Vol. no. 16**: 285–309, 1996.

[53] Mazoyer J. and Rapaport I. Inducing an order on cellular automata by a grouping operation. in *STACS'98*, LNCS **Vol. 1373**: 116–127, 1998.

[54] Mazoyer J. and Terrier V. Signals on one dimensional cellular automata. To appear in *Theoretical Computer Science.*, 1998.

[55] McCullough W. and Pitts W. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.*, **Vol. no. 5**: 115–133, 1943.

[56] Minsky M. *Finite and infinite machines*. Prentice Hall, 1967.

[57] Morita K. A Simple Construction Method of a Reversible Finite Automaton out of Fredkin Gates, and its Related Model. *Transactions of the IEICE*, **Vol. no. E**: 978–984, 1990.

[58] Morita K. and Imai K. A Simple Self-Reproducing Cellular Automaton with Shape-Encoding Mechanism. *Artificial Life V*, 1996.

[59] Moore E. F. Machine models of self-reproduction. *Proc. Symp. Appl. Math.* **Vol. no. 14**: 17–33, 1962.

[60] Mosconi J. La constitution de la théorie des automates. *Thèse d'Etat*, Université Paris 1, 1989.

[61] Myhill J. The converse of Moore's garden-of-eden theorem. *Proc. AMS* **Vol. no. 14**: 685–686, 1963.

[62] Peitgen H.-O., Rodenhausen A. and Skordev G. Self-similar Functions Generated by Cellular Automata. Research Report 426, Bremen University, 1998.

[63] Reimen N. Superposable Trellis Automata. *LNCS*,**Vol. no. 629**: 472–482, 1992.

[64] Rémila E. An introduction to automata on graphs. in *Cellular Automata: a parallel model*, Delorme M. and Mazoyer J. Eds, Kluwer: 345–352. To appear.

[65] Richardson D.Tesselations with local transformations. *Journal of Computer and System Sciences*, **Vol. no. 5**: 373–388, 1972.

[66] Rogers H. *Theory of Recursive Functions and Effective Computability*, MIT Press, 1967.

[67] Róka Zs. Simulations between Cellular Automata on Cayley Graphs. To appear in *Theoretical Computer Science*, 1998.

[68] Smith III A. Real-time language recognition by one-dimensional cellular automata. *Journal of Computer and System Sciences* **Vol. no. 6**: 233–253, 1972.

[69] Smith III A. Simple computation-universal spaces. *Journal of ACM.* **Vol. no. 18**: 339-353, 1971.

[70] Smith III A. Simple Non-trivial Self-Reproducing Machines. *Proceedings of the Second Artificial Life Workshop*, Santa Fe: 709–725, 1991.

[71] Sutner K. De Bruijn graphs and linear cellular automata. *Complex Systems* **Vol.5 no. 1**: 19–30, 1991.

[72] Sutner K. Linear cellular automata and De Bruijn automata. in *Cellular Automata: a parallel model*, Delorme M. and Mazoyer J. Eds, Kluwer: 303–319. To appear.

[73] Terrier V. Language recognizable in real time by cellular automata. *Complex systems* **Vol. no. 8**: 325–336, 1994.

[74] Terrier V. On real time one-way cellular automata. *Theoretical Computer Science* **Vol. no. 141**: 331–335, 1995.

[75] Thatcher J. Universality in the von Neumann cellular model. Tech. Report 03105-30-T, University of Michigan, 1964.

[76] Turing A. On computable numbers, with an application to the Entscheidungsproblem. *P. London Math. Soc.* **Vol. no. 42**: 230–265, 1936.

[77] von Neumann J. The General and Logical Theory of Automata. in *Collected Works, Vol. 5*, Taub. A., Eds, New York: Pergamon Press: 288–328, 1963.

[78] von Neumann J. *The Computer and the Brain*, Yale University Press, New Haven, 1958.

[79] von Neumann J. *Theory of Self-reproducing automata*, University of Illinois Press, Chicago, 1966.

[80] von Neumann J. *Probabilistic logic and the synthesis of reliable organisms from unreliable components*, Shannon and McCarthy Eds, Princeton University Press, Princeton, 1956.

[81] Wagner K. and Wechsung G. *Computational Complexity.* Reidel, 1986.

[82] Wiener N. *Cybernetics, or control and communication in the animal and the machine*, M.I.T. Press, New Haven, 1961.

[83] Willson S. Cellular automata can generate fractals. *Discrete Applied Mathematics*,**Vol. no. 8**: 91–99, 1984.

[84] Wolfram S. Twenty problems in the theory of cellular automata. *Physica Scripta* **Vol. no. 79**: 170–183, 1985.

[85] Wolfram S. *Theory and Applications of Cellular Automata*, World Scientific, 1986.