Atti Accademia Nazionale Lincei Classe Scienze Fisiche Matematiche Naturali

Rendiconti Lincei
Matematica e Applicazioni

Claudio Baiocchi

Some results on cellular automata

Atti della Accademia Nazionale dei Lincei. Classe di Scienze Fisiche, Matematiche e Naturali. Rendiconti Lincei. Matematica e Applicazioni, Serie 9, Vol. **9** (1998), n.4, p. 307–316.

Accademia Nazionale dei Lincei

<http://www.bdim.eu/item?id=RLIN_1998_9_9_4_307_0>

Atti della Accademia Nazionale dei Lincei. Classe di Scienze Fisiche, Matematiche e Naturali. Rendiconti Lincei. Matematica e Applicazioni, Accademia Nazionale dei Lincei, 1998.

Rend. Mat. Acc. Lincei
s. 9, v. 9:307-316 (1998)

Scienza dell'informazione. — *Some results on cellular automata.* Nota (*) del Corrisp. Claudio Baiocchi.

Abstract. — We want to discuss some properties of one-dimensional, radius 1, CUCAs (1). In particular, on one hand we want to keep small the number of states (2); on the other hand we are interested into automata, possibly requiring a high number of states, whose transition law is «as simple as possible»; *e.g. totalistic* automata (3). More generally, we will deal with the problem of simulating a generic cellular automaton through an automaton having a «simpler» transition law.

Key words: Cellular automata; Turing machine; Totalistic automata.

Riassunto. — *Qualche risultato sugli atomi cellulari.* Ci proponiamo di discutere qualche proprietà degli automi cellulari con capacità di calcolo universali, nell'àmbito di automi uni-dimensionali di raggio 1. Siamo in particolare interessati da un lato al problema di rendere basso il numero di stati, e d'altro lato ad automi che, sia pure con alto numero di stati, abbiano una legge di transizione particolarmente semplice. Più in generale, cercheremo di simulare un qualunque automa con uno la cui legge di transizione sia «più semplice».

1. Notations

We are given a finite set $\mathcal{S}$ (the set of the *states*) and a function $f : \mathcal{S}^3 \to \mathcal{S}$ (the *transition law*). Any (doubly infinite) sequence $\mathbf{c} \equiv (c_n)_{n\in\mathbb{Z}}$ with values $c_n \in \mathcal{S}$ will be called a *configuration*; the *future configuration* of $\mathbf{c}$ is the sequence $\widetilde{\mathbf{c}} \equiv (\widetilde{c}_n)_{n\in\mathbb{Z}}$ defined by $\widetilde{c}_n := f(c_{n-1}, c_n, c_{n+1})$; the couple $\{\mathcal{S}, f\}$ will be called *Cellular Automaton* (in short: *CA*).

Remark 1.1. Concerning possible generalizations we want to signal:

- *Bigger radius automata*: in the future configuration, the value of $\widetilde{c}_n$ could depend from $\left(c_{n+j}\right)_{|j|\le r}$. Here we will work only with radius $r = 1$.
- *Multi-dimensional automata*: *e.g.* the configurations could have the structure $\mathbf{c} \equiv$ $\equiv (c_{m,n})_{m,n\in\mathbb{Z}}$. For a suitable r, the value of $\widetilde{c}_{m,n}$ will now depend from $\left(c_{m+i,n+j}\right)_{\|(i,j)\|_p\le r}$. The most used norms correspond to $p = 1$ (Von Neumann's neighbourhood) and to $p = \infty$ (Moore's neighbourhood).

As a special case of two-dimensional automaton with Moore's neighbourhood and radius 1, let us recall the Conway's automaton «Life»: it has just two states (say $\mathcal{S} =$

(*) Pervenuta in forma definitiva all'Accademia il 27 luglio 1998.
(1) We denote by CUCA a Computationally Universal Cellular Automaton; see later on for the definitions.
(2) The first example of «small» CUCA is due to Smith III [13]; it requires 18 states.
(3) The existence of a totalistic CUCA, conjectured by Wolfram [14], was proved by Gordon [7] who constructed a totalistic CUCA with 9139 states.

{living, dead}) and a very simple transition law: a living cell will remain living iff it has 2 or 3 living neighbours; a dead cell will be brought to life iff it has exactly 3 living neighbours. Despite its simplicity, this automaton is in fact a CUCA [4, 6]; for an informal survey on the interesting properties of this automaton see [5]. □

We will often work with transition laws having a «simple» structure; *e.g.* with *symmetric* laws:

$$f(a, b, c) = f(c, b, a); \tag{1}$$

in the framework of $\mathcal{S} \subset \mathbb{N}$, the simplest symmetric law has the form:

$$f(a, b, c) = g(a + b + c) \text{ for a suitable } g : \mathbb{N} \to \mathbb{N}; \tag{2}$$

the corresponding automata (4) are called *totalistic*. When (as happens for *Life*) the future state of a cell depends not only from the sum of its neighbours but also from the cell itself, we will speak of *semi-totalistic* automata; in our framework this means that, for a suitable function $h : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, one has:

$$f(a, b, c) = h(b, a + c). \tag{3}$$

A *steady state* is an element $s \in \mathcal{S}$ such that $f(s, s, s) = s$; we will confine ourselves to the usual framework: one specifies a special steady state q, said the *quiescent state*, and one works only with *legal* configurations, say with sequences $\mathbf{c}$ such that $c_n = q$ for all indices n but a finite number (of course, for any legal $\mathbf{c}$, also $\widetilde{\mathbf{c}}$ is legal). When the set $\mathcal{S}$ is a subset of N, we will choose 0 as quiescent state (so that legal configurations are the compact supported ones) and we will impose:

$$\begin{cases} g(0) = 0 & \text{in the framework of (2);} \\ h(0, 0) = 0 & \text{in the framework of (3);} \\ f(0, 0, 0) = 0 & \text{in the general case.} \end{cases} \tag{4}$$

2. Simulations of Turing Machines

The Cellular Automata are a quite powerful device; *e.g.* the instantaneous description of a $TM[n, m]$ (Turing Machine with n internal-states and m tape-symbols) can be viewed as a CA (5); in particular the simulation of a UTM (*Universal Turing Machine*) gives rise to a CUCA.

Remark 2.1. In a Turing Machine the whole power is concentrated on the scanned cell; on the contrary, in Cellular Automata, any cell has its own power; in particular

(4) Deeply investigated by Wolfram in [14].

(5) If the internal states and the tape-symbols vary respectively in $\{1, \dots, n\}$ and $\{0, \dots, m-1\}$, we can choose $\mathcal{S} = \{0, \dots, m-1\} \times \{0, \dots, n\}$: any non-scanned cell containing the tape-symbol x will be coded by $(x, 0)$; for $y \in \{1, \dots, n\}$ the code (x, y) will denote the Head in state y scanning the tape-symbol x.

one can expect a better efficiency of Cellular Automata; such a conjecture is in fact validated by many examples, *e.g.* the *firing squad syncronization problem* (see [3]). On the other hand all examples of CUCA we will consider are obtained by simulating a UTM; so that the efficiency will not increase ... □

Depending on the type of simulation, the number of states for a CA which simulates a TM can take various forms; let us collect some results in this direction:

THEOREM 2.1. *Any* $TM[n, m]$ *can be simulated by a k-states CA where*:
1. $k = (n + 1) \cdot m$
2. $k = m + 2n$
3. $k = m + n + 4$
4. $k = m + n + 2$
5. $k = \max[m, n] + 4$

the simulations 4 *and* 5 *giving raise to illegal CA.* □

The first result follows obviously from footnote 5. The second one is the well known result of Smith III [13]; the key idea is that both the Head of the machine must have *their own* representing cell into the CA; concerning the states of the Automaton, m states will code the tape-symbols; n states will code the Head «looking right»; the remaining n will code the Head «looking left».

Following [2], let us now use *three* contiguous CA's cells to code the Head; the central one being the «true Head», and the flanking ones having a twofold task: on one hand they act as a *modulo three clock*, on the other hand they «drive» the Head in its work. Four new-states are sufficient to code the drivers; in such a way one gets the formula $k = m + n + 4$ (for the details see [2]).

Formula 4, due to [8], is based on a different idea: in the simulating Automaton the tape-cells are intermixed with another type of cells, acting as a modulo-two clock; two new values are sufficient in order to realize the simulation. Of course the simulating CA is illegal: near $\pm\infty$ the configuration is periodic (both in space and in time).

The last result, due to Goles, Margenstern and Matamala, is quoted in [10, p. 144], as *to appear*. □

Starting from Theorem 2.1, the simulation of a UTM will fournish examples of CUCAs. *E.g.*, simulating the UTM [7,4] constructed by Minski [10], formula $k = m + 2n$ gives the 18-states CUCA quoted in footnote 2.

Let us point out that the Minski's machine was for a while the «smallest» known UTM, and most of the small CUCAs was constructed through it; of course smaller UTMs can give raise to smaller CUCAs. A lot of small UTMs have been constructed by Rogozhin (see [11]); in particular, the Rogozhin's UTM[5,5] gives a 14-states UTM (9 states in the illegal framework).

REMARK 2.2. As pointed out by Lindgren and Nordahl [8], an explicit use of the TM's Table can give raise to smaller CUCAs; let us summarize some results of [8].

- Denoting by n_R (resp. n_L) the number of states the head can assume when it must go Right (resp. Left), the Smith's formula $k = m + 2n$ can be sharpened into $k = m + n_R + n_L$ (6).
- Further details on the Table can give raise to even smaller CUCAs; *e.g.*, using a composite (two-cells) object to represent the Head, one can simulate the Minski's UTM[7,4] through a 7-states CUCA (illegal, because it requires a periodic background).
- In order to have a legal simulation, two more states suffice (7).

Concerning the use of the whole Table's structure, we do not know any attempt using the small Rogozhin's UTMs. □

Remark 2.3. A somewhat different idea giving rise to a small CUCA is due to Albert and Culik II (see [1]); it uses only 14 states and, instead of simulating a universal TM, works as a universal CA: taking as input a transition law and a starting configuration of an automaton $\mathcal{A}$, it simulates the evolution in $\mathcal{A}$ of such a configuration. □

3. Simulations through special laws

Concerning semi-totalistic simulations, we proved in [2] the following result:

Theorem 3.1. *Any k-states CA can be simulated by a k^2-states semi totalistic CA.*

Proof. Let us choose $\mathcal{S} = \{0, \dots, k-1\}$ for the original CA; the simulating one will have $\hat{\mathcal{S}} = \{0, \dots, k^2 - 1\}$. We construct our simulation by replacing any sequence with values in $\mathcal{S}$ by a suitably chosen sequence with values in $\hat{\mathcal{S}}$; precisely:

$$(5) \qquad \begin{cases} \text{we replace:} & \cdots\cdots & a & b & c & d & \cdots\cdots \\ & & & & & & \\ \text{with:} & \cdots\cdots & a+kb & b+kc & c+kd & d+\cdot & \cdots\cdots \end{cases}$$

We claim that (the value of k being fixed):

$$(6) \qquad \begin{cases} \text{from the knowledge of } b+kc \text{ and } (a+kb)+(c+kd), \\ \text{one can reconstruct the four values } a, b, c, d. \end{cases}$$

The proof is quite elementary: because of $a, b, c, d \in \{0, \dots, k-1\}$, from $b + kc$ we can evaluate b, c; we then subtract $c + kb$ from the sum, thus getting $a + kd$; this last quantity obviously fournishes a, d. Thus, given *any* transition law f acting on the original sequence $(\cdots a, b, c, d, \cdots)$, we can follow the evolution by operating with a semi-totalistic law on the replacing sequence $(\cdots, a+kb, b+kc, c+kd, d+ke, \cdots)$. In other words: knowing a (new-type) cell $b + kc$ and the sum of the flanking ones $(a + kb) + (c + kd)$, due to (6) we can evaluate $f(a, b, c) + kf(b, c, d)$, say the future state of the central cell. □

(6) The proof is immediate; thus, simulating the Minski's UTM[7,4] and the Rogozhin UTM[5,5], one easily gets a legal CUCA with respectively 13 and 12 states.

(7) Further details on this point will be given in the last section.

REMARK 3.1. The second row in (5) could appear more expressive if written in the form:

$$\cdots\cdots \binom{a}{b}\binom{b}{c}\binom{c}{d} \cdots\cdots$$

In the following we will often use a *vector-valued* set $\mathcal{S}$; remark that for states in $\mathbb{N}^p$ the notion of (semi) totalistic law makes sense; the translation of vectors into numbers is just matter of avoiding carries. □

REMARK 3.2. Let $s \in \mathcal{S}$ be a steady state for the starting CA; then the vector $\binom{s}{s}$ is a steady state for the simulating CA. In particular if, as we did in (4), we code by 0 the quiescent state in the starting Automaton, our simulation is *legal*: any compact supported sequence is transformed into a sequence which is again compact supported. □

REMARK 3.3. Theorem 3.1 has a very short and elementary proof; in particular the simulation of the (simple) 12-states CUCA quoted in footnote 6 gives an easy proof of the existence of a 144-states semi-totalistic CUCA (8) By using more tricky CUCAs (*e.g.* the illegal 7-states of [8], or the legal 8-states we will construct in the last section) the number 144 can be laid down ... □

REMARK 3.4. Let us change the definition (5) by setting (with vector notations):

$$(7)\qquad \begin{cases} \text{we replace}: & \cdots\cdots & a & b & c & d & \cdots\cdots \\ \text{with}: & \cdots\cdots & \binom{a}{b} & \binom{c}{d} & \binom{d}{\cdot} & \binom{\cdot}{\cdot} & \cdots\cdots \end{cases}$$

We can follow the evolution of the starting Automaton by using, in the simulating one, a *one-way* transition law (each cell knows only its own state and the state of the cell on its right): denoting by f the original law, we define:

$$f_\star\left(\binom{a}{b}, \binom{c}{d}\right) := \binom{f(a, b, c)}{f(b, c, d)}$$

(thus $\binom{q}{q}$ will be the new quiescent state). Though affected by a shift (9), the evolution in the new CA allows to follow the evolution in the old one. The universal CA quoted in Remark 2.3 has the one-way simulation as one of its ingredients. □

By using some more sofisticated arguments, one can prove (see [2]):

THEOREM 3.2. *Any symmetric k-states CA can be simulated by a totalistic CA with* $O(k^4)$ *states*; *the simulation is such that, starting from a legal CA, also the simulating Automaton is legal.* □

By combining Theorems 3.1 and 3.2 one gets that any k-states CA can be simulated by a totalistic CA with $O(k^8)$ states. However, in the framework of *legal* Automata one can reach a better bound. In order to do that, we will need the following lemma:

(8) The first example in this framework, constructed by Gordon [7], requires 967 states.

(9) *E.g.*, for the trivial law $f(u, v, w) = v$, one has that $f_\star^2$ is the left-shift.

LEMMA 3.1. *We are given a vector $\overrightarrow{V}$ such that, for suitably chosen (unknown) values $a, b, c, d, e \in \mathbb{N}$, has one of the forms:*

$$\overrightarrow{V} \equiv \begin{pmatrix} b \\ c \\ d \\ a+c \\ b+d \\ c+e \end{pmatrix} \qquad \text{or} \qquad \overrightarrow{V} \equiv \begin{pmatrix} a+c \\ b+d \\ c+e \\ b \\ c \\ d \end{pmatrix}.$$

Then we can reconstruct the five values a, b, c, d, e; furthermore, if $\overrightarrow{V} \neq \overrightarrow{0}$, we can establish which was the form of $\overrightarrow{V}$.

PROOF. We denote by v_j $(j = 1, \dots, 6)$ the components of $\overrightarrow{V}$; of course, if we know which is the form of $\overrightarrow{V}$, from the components v_j we easily get a, b, c, d, e.

In order to discover which is the form of $\overrightarrow{V}$ we remark that, when $\overrightarrow{V}$ has the first form, one has $v_1 + v_2 + v_3 \leq v_4 + v_5 + v_6$; the equality being possible only when $a = c = e = 0$ (and in particular $v_2 = 0$). On the contrary, when $\overrightarrow{V}$ has the second form, one has $v_1 + v_2 + v_3 \geq v_4 + v_5 + v_6$; the equality being possible only when $a = c = e = 0$ (and in particular $v_5 = 0$). The only possible doubt corresponds to $a = c = e = 0$ and $v_2 = v_5 = 0$; these last relations imply, for both forms of $\overrightarrow{V}$, that $b = d = 0$, say $\overrightarrow{V} = \overrightarrow{0}$. $\square$

Now let us come back to totalistic simulations. In the simulating CA, each cell will know the values of *three* old cells (instead of two as in (5)). By using vector-valued states (see Remark 3.1) we will replace the generic configuration $\mathbf{c} \equiv (c_n)_{n\in\mathbb{Z}}$ by the configuration $\mathbf{C} \equiv (C_n)_{n\in\mathbb{Z}}$ where (10):

$$C_n := \begin{pmatrix} c_{n-1} \\ c_n \\ c_{n+1} \\ 0 \\ 0 \\ 0 \end{pmatrix} \text{ if } n \text{ is odd;} \qquad C_n := \begin{pmatrix} 0 \\ 0 \\ 0 \\ c_{n-1} \\ c_n \\ c_{n+1} \end{pmatrix} \text{ if } n \text{ is even.} \tag{8}$$

For the sum $C_{n-1} + C_n + C_{n+1}$ (which involves 5 «old» values, say a, b, c, d, e) we will have that, if n is odd:

$$C_{n-1} + C_n + C_{n+1} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ a \\ b \\ c \end{pmatrix} + \begin{pmatrix} b \\ c \\ d \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ c \\ d \\ e \end{pmatrix} = \begin{pmatrix} b \\ c \\ d \\ a+c \\ b+d \\ c+e \end{pmatrix}$$

(10) Remark that, starting from a legal $\{c_n\}_{n\in\mathbb{Z}}$, the corresponding $\{C_n\}_{n\in\mathbb{Z}}$ is also legal.

while, if n is even:

$$C_{n-1} + C_n + C_{n+1} = \begin{pmatrix} a \\ b \\ c \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ b \\ c \\ d \end{pmatrix} + \begin{pmatrix} c \\ d \\ e \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} a+c \\ b+d \\ c+e \\ b \\ c \\ d \end{pmatrix}.$$

Concerning the future value $\widetilde{C}_n$ of the central cell C_n, let us define $\widetilde{b}, \widetilde{c}, \widetilde{d}$ by means of $\widetilde{b} := f(a, b, c)$, $\widetilde{c} := f(b, c, d)$, $\widetilde{d} := f(c, d, e)$; then one has:

$$(9) \qquad \widetilde{C}_n = \begin{pmatrix} \widetilde{b} \\ \widetilde{c} \\ \widetilde{d} \\ 0 \\ 0 \\ 0 \end{pmatrix} \text{ if } n \text{ is odd;} \qquad \widetilde{C}_n = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \widetilde{b} \\ \widetilde{c} \\ \widetilde{d} \end{pmatrix} \text{ if } n \text{ is even.}$$

In order to achieve a totalistic simulation, from the sum $C_{n-1} + C_n + C_{n+1}$ we need to reconstruct $\widetilde{C}_n$. Of course, in order to evaluate $\widetilde{b}, \widetilde{c}, \widetilde{d}$, the knowledge of a, b, c, d, e suffices; but, in order to construct $\widetilde{C}_n$, *we also need to know if n is odd or even*! However, in the framework of *legal* Automata, we do not need to know n: we just apply Lemma 3.1 to the vector $\overrightarrow{V} := C_{n-1} + C_n + C_{n+1}$. If such a $\overrightarrow{V}$ does not vanish, we can reconstruct its «type», so we know which form is needed for $\widetilde{C}_n$; otherwise it was $a = b = c = d = e = 0$, and in such a case we simply set $\widetilde{C}_n := \overrightarrow{0}$.

By an obvious «numerization» of the vectors C_n, and taking into account the remark in footnote 10, we thus proved:

THEOREM 3.3. *In the framework of legal Automata, any k-states CA can be simulated by a totalistic $O(k^6)$-states CA.* □

REMARK 3.5. We showed in [2] that Lemma 3.1 can easily be adapted to multi-dimensional cases. In particular the simulation of *Life* gives:

$$(10) \qquad \begin{cases} \text{in dimension 2, with the Moore's neighbourhood,} \\ \text{there exists a radius-1 totalistic CUCA.} \end{cases}$$

A similar result for the neighbourhood of Von Neumann is, as far as we know, an open problem. □

The bound $O(k^6)$ in Theorem 3.3 can seem very high; let us develop some remarks about it.

- For a K-states totalistic Automaton there exist K^{3K-3} transition laws (something less in the legal framework); thus, in order to simulate any k-states CA (11), we need at least $K = O(k^3)$.

(11) Say: k^{k^3} transition laws; something less in the legal framework.

- In [2] we proved a variant of Lemma 3.1 using vectors with only 5 components; this enforces Theorem 3.3, requiring just $O(k^5)$ states.
- A technique of «cyclically coloured cells» proposed in [1] gives the bound $O(k^4)$. It does not respect the legal framework, but has the advantage of using just $4k$ values for the simulating cells.
- The optimal bound $O(k^3)$ can easily be reached; see [2]. It is an illegal simulation, again based on colouring cells; a *legal* $O(k^3)$ simulation can be obtained (see [2, Osservazione 3.2]) by adding a «dirty trick» we will detail, in a different framework, in the last section.

We thus have a new proof of the Wolfram's conjecture about the existence of totalistic CUCAs: it is sufficient to simulate any UTM (or any CUCA). However the situation is quite different with respect to the semi-totalistic case (see Remark 3.3): the existence proof for the simulating automaton is still easy, but the number of states required for the simulation is very high! It is more convenient to use Theorem 3.2, which gives a better bound with respect to Theorem 3.3; of course the part of the tape close to the head (where the transition law is no longer symmetric) requires a special treatment. The interested reader can find the details in [2]; let us just recall that, when m is a power of 2, one has:

$$\text{(11)} \qquad \begin{cases} \text{any } TM[n, m] \text{ can be simulated through} \\ \text{a totalistic CUCA with } \dfrac{1 + 8n + (12n + 2)(m^4 - 1)}{3} \text{ states} \end{cases}$$

so that the simulation of the UTM[24,2] of Rogozhin [11] gives:

(12) there exists a 1663-states totalistic legal CUCA.

Remark 3.6. We recall that, accordingly with our conventions, formula (12) means that the maximum state-value used in the CUCA is 1662; but in fact few values between 0 and 1662 are used by the simulation. When measuring the Automaton's size in terms of the *used values* (instead of in terms of the *maximum value*), and accepting illegal simulations, smaller totalistic CUCA can be obtained by using the already quoted colouring-cells strategy suggested in [1]. □

4. A legal 8-states CUCA

We now come back to the illegal 7-states CUCA of [8] quoted in Remark 2.2. Two of the 7 states, say α, β (12) satisfy:

$$\text{table 1:} \qquad \begin{cases} f(\alpha, \alpha, \alpha) = \alpha; f(\beta, \beta, \beta) = \beta \\ f(\alpha, \beta, \beta) = f(\beta, \beta, \alpha) = \beta; f(\beta, \alpha, \beta) = \alpha \end{cases}$$

In particular both α, β are stable; and also stable is the (illegal) configuration **b**, periodic

(12) Denoted 0 and ⊔ in [8].

of period three, given by:

$$\mathbf{b} := \cdots\cdots \beta\alpha\beta\beta\alpha\beta\beta\alpha\beta \cdots\cdots$$

The rules in this CA are constructed in such a way that it can simulate the TM[7,4] of Minski; however, in order to realize such a simulation, the «interesting part» of the tape must be embedded in the periodic background **b**; thus loosing the legal framework.

In order to have a legal simulation, in [8] is asserted that two more states suffice; let us show that:

(13) just one further state is needed.

Precisely, fix any «new state» $\gamma \notin \mathcal{S}$; set $\mathcal{S}\star := \mathcal{S} \bigcup \{\gamma\}$ and extend the transition law f into a law $f_\star$ by setting:

$$\text{table 2:} \qquad \begin{cases} f_\star(\gamma\,,\gamma\,,\alpha) = f_\star(\alpha\,,\gamma\,,\gamma) = \alpha \\ f_\star(\alpha\,,\gamma\,,\alpha) = f_\star(\beta\,,\gamma\,,\gamma) = f_\star(\gamma\,,\gamma\,,\beta) = \beta \\ f_\star(\gamma\,,\alpha\,,\alpha) = f_\star(\alpha\,,\alpha\,,\gamma) = \gamma \end{cases}$$

and completing the definition by means of the trivial law (13). Let us follow three steps in the evolution of a special finite configuration (14):

$$\begin{cases} \text{from:} & \beta & \alpha & \beta & \gamma & \gamma & \alpha & \alpha & \alpha & \alpha & \alpha & \alpha \\ \text{one gets:} & ? & \alpha & \beta & \beta & \alpha & \gamma & \alpha & \alpha & \alpha & \alpha & ? \\ \text{then:} & ? & ? & \beta & \beta & \alpha & \beta & \gamma & \alpha & \alpha & ? & ? \\ \text{and then:} & ? & ? & ? & \beta & \alpha & \beta & \gamma & \gamma & ? & ? & ? \end{cases}$$

so that, in three steps, the block $\beta\alpha\beta\gamma\gamma$ followed by all α moves three cells right, no matter what is on its left. Because of the symmetry of tables 1 and 2, one has also the mirror property: the block $\gamma\gamma\beta\alpha\beta$, when has all α on its left, moves three cells left in three steps of the Automaton.

For any finite sequence $\mathcal{F}$, in order to follow the evolution the illegal configuration:

$$\cdots\beta\alpha\beta\cdots\beta\alpha\beta\mathcal{F}\beta\alpha\beta\cdots\beta\alpha\beta\cdots$$

is thus sufficient to follow the evolution of the legal (15) configuration:

$$\cdots\alpha\alpha\cdots\alpha\alpha(\gamma\gamma\beta\alpha\beta)\mathcal{F}(\beta\alpha\beta\gamma\gamma)\alpha\alpha\cdots\alpha\alpha\cdots$$

where the inserted moving blocks create the background needed by $\mathcal{F}$ in order to properly work; assertion (13) is thus proved.

Remark 4.1. As already remarked, either α or β could be chosen as quiescent state. If we want to work with β as quiescent state, we define the nontrivial part of $f_\star$ by

(13) Say: for the remaining triples $(x, y, z) \in \mathcal{S}_\star \times \mathcal{S}_\star \times \mathcal{S}_\star - \mathcal{S} \times \mathcal{S} \times \mathcal{S}$ we set $f_\star(x, y, z) = y$.
(14) The states on the left and on the right, as well as the ? that they generate, are irrelevant.
(15) We assume here that α is the quiescent element; see Remark 4.1 later on.

replacing table 2 with:

$$\begin{cases} f_\star(\gamma\,,\gamma\,,\gamma)=\alpha \\ f_\star(\alpha\,,\gamma\,,\gamma)=f_\star(\gamma\,,\gamma\,,\alpha)=f_\star(\beta\,,\gamma\,,\alpha)=f_\star(\alpha\,,\gamma\,,\beta)=\beta \\ f_\star(\beta\,,\beta\,,\gamma)=f_\star(\gamma\,,\beta\,,\beta)=\gamma \end{cases}$$

Now the evolution of the special block followed by β are:

$$\begin{cases} \text{from:} & \beta & \alpha & \beta & \gamma & \gamma & \beta & \beta & \beta & \beta & \beta & \beta \\ \text{one gets:} & ? & \alpha & \beta & \gamma & \gamma & \gamma & \beta & \beta & \beta & \beta & ? \\ \text{then:} & ? & ? & \beta & \gamma & \alpha & \gamma & \gamma & \beta & \beta & ? & ? \\ \text{and then:} & ? & ? & ? & \beta & \alpha & \beta & \gamma & \gamma & ? & ? & ? \end{cases}$$

thus the needed background for finite configurations $\mathcal{F}$ is now created by the halflines $(\beta)^\infty\gamma\gamma\beta\alpha\beta$ on the left of $\mathcal{F}$ and $\beta\alpha\beta\gamma\gamma(\beta)^\infty$ on the right. □

Some results here described were presented during a conference held at the Istituto Lombardo, Milan 23-10-97 and will appear in [2].

References

[1] J. Albert - K. Culik II, *A Simple Universal Cellular Automaton and its One-Way and Totalistic Version*. Complex Systems, 1, 1987, 1-16.

[2] C. Baiocchi, *Qualche risultato sugli automi cellulari*. Rend. Ist. Lombardo, to appear.

[3] R. Balzer, *An 8-state minimal time solution to the firing squad sincronization problem*. Information and Control, 10, 1967, 22-42.

[4] E. R. Berlekamp - J. H. Conway - R. K. Guy, *Winning Ways for Your Mathematical Plays*. Academic Press, London 1982.

[5] A. K. Dewdney, *Computer recreations*. Sci. Amer., 224, 1971, 226, 1972.

[6] M. Gardner, *Wheels, Life and other mathematical amusements*. Freeman, 1983.

[7] D. Gordon, *On the Computational Power of Totalistic Cellular Automata*. Math. Systems Theory, 20, 1987, 43-52.

[8] K. Lindgren - M. G. Nordahl, *Universal Computation in Simple One-Dimensional Cellular Automata*. Complex Systems, 4, 1990, 299-318.

[9] M. Margenstern, *Frontier between decidability and undecidability: a survey.* In: M. Margenstern (ed.), *Proceedings of MCU'98*. Vol. II, Metz 1998, 141-177.

[10] M. Minski, *Computation: Finite and Infinite machines*. Prentice Hall, 1967.

[11] Ju. V. Rogozhin, *Small universal Turing machines*. Theor. Comp. Sci. 168-2, 1987.

[12] C. E. Shannon, *A Universal Turing machine With Two Internal States*. In: C. E. Shannon - J. McCarthy (eds.), *Automata Studies*. Princeton University Press, 1956.

[13] A. R. Smith III, *Simple Computational-Universal Cellular Spaces*. J. of ACM, 18, 1971.

[14] S. Wolfram, *Statistical Mechanics of Cellular Automata*. Rev. Modern Physics, 55, 1983, 601-644.

[15] S. Wolfram (ed.), *Theory and Application of Cellular Automata*. Advanced series on complex systems, 1, World Scientific, 1987.

Pervenuta il 13 marzo 1998,
in forma definitiva il 27 luglio 1998.

Dipartimento di Matematica
Università degli Studi di Roma «La Sapienza»
Piazzale A. Moro, 5 - 00185 Roma
baiocchi@mat.uniroma1.it