



# A particular universal cellular automaton

Gaétan Richard, Nicolas Ollinger

## ► To cite this version:

| Gaétan Richard, Nicolas Ollinger. A particular universal cellular automaton. 2006. hal-00095821v1

**HAL Id: hal-00095821**

**<https://hal.science/hal-00095821v1>**

Preprint submitted on 18 Sep 2006 (v1), last revised 7 Apr 2008 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A particular universal cellular automaton

Gaétan RICHARD

LIF, CNRS & Université de Provence  
39, rue Joliot-Curie  
13453 Marseille Cedex 13  
`gaetan.richard@lif.univ-mrs.fr`

**Abstract.** We give a new small intrinsically universal cellular automaton with 4 states (improving the best known result). The construction takes heavy use of self-organisation: from simples structures – particles and collisions –, we build a complex construction able of simulating any other cellular automaton.

Complex systems are dynamical systems characterised by a great number of simple base elements. Among those systems, some exhibits interesting behaviors: on some cases, the “complexity” of the whole system seems far greater than just the sum of complexity of elementary cells (such a phenomenon is called *emergence*). On some other cases, even if the local behavior seems random, on a macroscopic scale one can observe regular and predictable structures. This behavior is referred as *self-organisation*. To be able to conduct a theoretical study of these phenomena, one must have a formal model and some examples (the smaller, the better) of systems that exhibit these behaviors. A possible approach is to use cellular automata, which are dynamical systems in discrete time and space. This paper’s aim is to present a small intrinsically universal<sup>1</sup> cellular automaton.

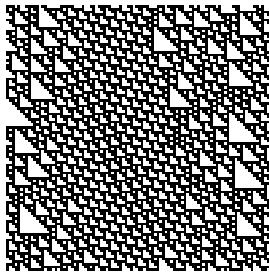
In this paper, a cellular automaton will refer to a pair  $(S, f)$ , where  $S$  is a finite set of *states* and  $f : S^3 \rightarrow S$  is called *local rule*. Cellular automata are dynamical systems on *configurations* (which consists on a mapping  $c : \mathbb{Z} \rightarrow S$ ) according to the global rule  $G$  defined by  $G(c)(i) = f(c(i-1), c(i), c(i+1))$ . We will say that a cellular automaton is one-way if the local rule does not depend on the last variable. In a more general context, cellular automata can use configurations on  $\mathbb{Z}^d$  (in this case,  $d$  is called dimension) and with a local rule with a greater arity (noted  $v$ ). A survey of known results on cellular automaton was written by J. Kari [4].

To observe the dynamic of cellular automaton, one convenient representation is *space-time diagram* which consist in pushing all configuration in one orbit. . Formally, for a given configuration  $c_0$ , the space-time diagram generated by  $c_0$  is the application  $D : \mathbb{Z} \times \mathbb{N} \rightarrow S$  with  $D(x)(y) = G^y(c_0)(x)$ . An example of such a diagram is depicted on figure 1. In this picture a self-organisation phenomenon

---

<sup>1</sup> i.e. who is able to simulate any other automaton (formal definition will be given in the article)

occur: simple structures – periodic patterns – appear and interact with each other. The aim of this paper is to provide a small automaton that exhibits such a behavior.



**Fig. 1.** Extract of a space-time diagram

To quantify the range of possible behaviors of a cellular automaton, a method taken from complexity theory is to compare with a reference model: the Turing Machine. Even if both system are in discrete time and discrete space, they are have strong differences since Turing machine works only on finite words and outputs a result contrary to cellular automaton. However, with some adaptation, it can be proved that some cellular automata have at least as much computational power as Turing machine. Such automata are called *Turing universal*. For a formal definition, we will use the one presented by B. Durand and Z. Róka (see [3]): An automaton  $\mathcal{A}$  is said *Turing Universal* if, for any Turing Machine  $M_T$ , there exist an halting motif  $e$  (finite sub-configuration of  $\mathcal{A}$ ) and a LOG-SPACE transformation from any (finite) configuration  $s_0$  of the Turing Machine to an ultimately periodic configuration  $c = c_g^\omega c_0 c_d^\omega$  of  $\mathcal{A}$  such that  $M_t$  halts on  $s_0$  if and only if  $e$  appears in the evolution of  $\mathcal{A}$  with starting configuration  $c$ .

Turing universal automata are able of arbitrary complex behavior on finite subset of the space time diagram. However, complex systems deals mostly with infinite configurations which limits the interest of Turing universality.

To overcome this problem, one can have a new kind of universality: *intrinsic universality*. The main idea for this universality is to simulate any other automaton preserving its dynamics. A possible implementation is to impose the existence of a local geometrical transformation between space-time diagram.

A way to formalise such a definition is the following: an automaton  $B = (S_B, f_B)$  simulates an automaton  $A = (S_A, f_A)$  if there exists an one-to-one morphism  $f : S_A \rightarrow S_B^k$  and two integers  $n_A$  and  $n_B$  such as for all  $c_a$  configuration of  $A$ , we have the diagram given in figure 2. For a study on this order, the reader may look at N. Ollinger thesis [8]. An automaton will be said *intrinsically universal* if it can simulate any other automaton. One can note that this definition is strictly stronger than Turing universality: every intrinsically universal automaton is Turing universal but there exists Turing universal automata that are not intrinsically universal.

$$\begin{array}{ccc}
c'_A & \xrightarrow{f} & c'_B \\
G_A^{m_A} \uparrow & & \uparrow G_B^{m_B} \\
c_A & \xrightarrow{f} & c_B
\end{array}$$

**Fig. 2.** Simulation between automata

For our particular case, the choice has been to relax a little the definition by transforming the constraint  $f$  one-to-one into  $f$  is a composition of an injective function and the inverse of a surjective one. This definition is compatible with B. Durand and Z. Róka formalisation (see [3]). The main reason behind this choice is concision: the result can be proved with the more restrictive one, however at the price of a very verbose and long proof that does not add any interest. Moreover, it is currently an open question to know whether those definitions are equivalent or not.

With those definitions, it is now possible to give formal results on small universal automata. The table 1 present some historical result. Note that it was proved by N. Ollinger that the problem of knowing if an automaton is intrinsically universal is undecidable (see [10]). This explain that the search was mainly made by hand. For our kind of automata, the best known result of intrinsically universal automata is N. Ollinger 6 states' one. For Turing universality, M. Cook and S. Wolfram have a better and optimal result with only 2 states (see [2]). This automaton is proved to be Turing universal. Even more, it was recently shown by T. Neary and D. Woods that checking a partial space-time diagram is P-complete [6].

However, those results do not prove that this automaton is intrinsically universal.

Year	author	d	v	states	type
1966	J. Von Neumann [7]	2	5	29	intrinsic
1970	E. R. Banks [1]	2	9	2	intrinsic
		1	3	18	intrinsic
		1	5	2	intrinsic
1990	K. Lindgren, M. G. Nordhal [5]	1	3	7	Turing
2002	N. Ollinger [9]	1	3	6	intrinsic
2004	M. Cook, S. Wolfram [2]	1	3	2	Turing

**Table 1.** Some known results about small automata

The main aim of this paper is to improve the bound on the minimal size of an intrinsically universal cellular automaton.

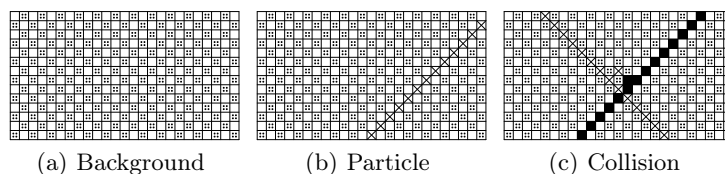


## 1 Particles and collisions

To construct small automata, a common way is to use particles and collisions, this is also the case in our construction. Before entering the core of the proof, some explanations will be given on those objects.

Particles and collisions are useful tools when it comes to construct small and complex automaton. For example, one can cite the (informal) definition of class 4 of S.Wolfram (see [11]): *In most cases, all sites are seen to “die” (attain value zero) after a finite time. However, in a few cases, stable or periodic structures which persist for an infinite time are formed. An important feature (...) is the presence of propagating structures. By arranging for suitable reflections of these propagating structures, final states with any cycle lengths may be obtained.*

Among the structures found in most of constructions, it is possible to distinguish three main objects: First, a *background* (see fig. 3(a)) is a regular structure that “fills” wide zone of the space-time diagram. Inside backgrounds, stable periodic structures will be called *particles* (see fig. 3(b)). Particles last forever unless they collide into another particle. The generated element is called *collision* (see fig. 3(c)). Any collision takes a set of particles as input and outputs another (possibly empty) set.



**Fig. 3.** Example of elements

Of course, it is possible to give formal definitions of particles and collisions. It is even possible to build formal tools to manipulate these objects and check their validity. We choose to not use this kind of formal approach in this paper since the overhead needed is way to big for this case where all uses of particles and collisions are simple. This approach may be however useful and even necessary in more complex cases.

## 2 Constructing a 4 states automaton

The rest of the article will be dedicated to present a new small intrinsically universal automaton. Its main property is described in the following theorem:

**Theorem 1.** *The 4 states automaton whose local rule is given in table 2 is intrinsically universal.*

## 2.1 Local rule and global view

This automaton has a set of states  $S = \{L, R, 1, 2\}$ . The local rule is given in table 2. To show the intrinsically universality, we prove that this automaton can simulate any one-way cellular automaton.

$\frac{L}{L} \mid \frac{L \ R \ 1 \ 2}{L \ L \ R \ 2 \ R}$	$\frac{R}{L} \mid \frac{L \ R \ 1 \ 2}{L \ L \ 1 \ 2}$
$R \ R \ R \ R \ R$	$R \ L \ L \ 2 \ 1$
$1 \mid 1 \ 1 \ - \ R$	$1 \mid L \ L \ L \ 1$
$2 \mid R \ 2 \ L \ -$	$2 \mid L \ 2 \ 2 \ 2$

$\frac{1}{L} \mid \frac{L \ R \ 1 \ 2}{L \ L \ 2 \ 1 \ R}$	$\frac{2}{L} \mid \frac{L \ R \ 1 \ 2}{L \ L \ 1 \ 1 \ 2}$
$R \ 1 \ R \ R \ R$	$R \ 2 \ R \ R \ R$
$1 \mid L \ 1 \ R \ R$	$1 \mid 2 \ R \ L \ 1$
$2 \mid - \ 2 \ L \ 1$	$2 \mid L \ 2 \ 2 \ 1$

**Table 2.** Local transition rule of the 4 state automaton

the value of  $f(l, c, r)$  is given on the row  $l$  and column  $r$  of tabular  $c$

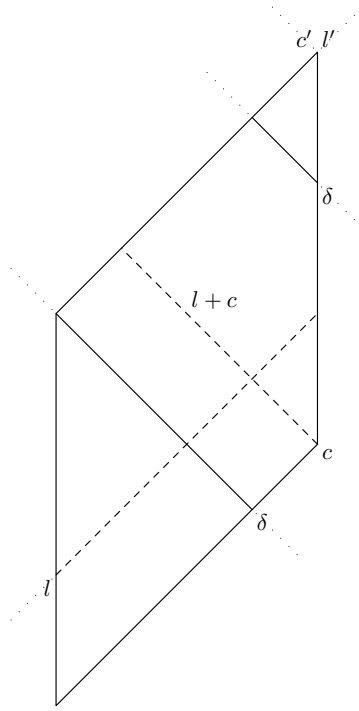
All simulations are based on the use of a unique widget, that will be called the widget in the paper. It can be represented by the symbolic drawing given in figure 4. This widget can be seen as taking three inputs  $\delta$ ,  $l$  and  $c$  and delivering three outputs  $\delta, l', c'$  ( $\delta$  is left unmodified). One can see that it is possible to tile the plan with widget.

## 2.2 Widget implementation

The main goal of this section is to implement the widget with the automaton. Doing this will fix the relations between the inputs and outputs of the widget. The macroscopic symbolic representation is given in figure 5. In this figure, each line represent a particle: the gray areas indicate place where more particles can be added. Figure 6 gives microscopic details of the widget.

Let  $N$  be a integer. Inside the widget, the signal  $\delta$  is encoded by using  $N + 1$  parallels signals (see for example area 1). Information will be encoded in the size of the gap between two consecutive particles and noted  $(\delta_i)_{0 \leq i < N}$  (this size correspond to the number of cells in state  $R$  or 1 between two signals). Signals  $l$ ,  $c$ ,  $l'$  and  $c'$  are implemented by two particles. They convey a value (smaller than  $N$ ) encoded with the size of the gap between the two particles (taken as the number of cells in state 2).

In the microscopic world, widget are vertically delimited by a vertical particle (see for example area 3). Once entering the widget, signal  $l$  cross the function  $\delta$  without any change (see area 1). Then the signal  $l$  encounter the signal  $c$  (area 2).



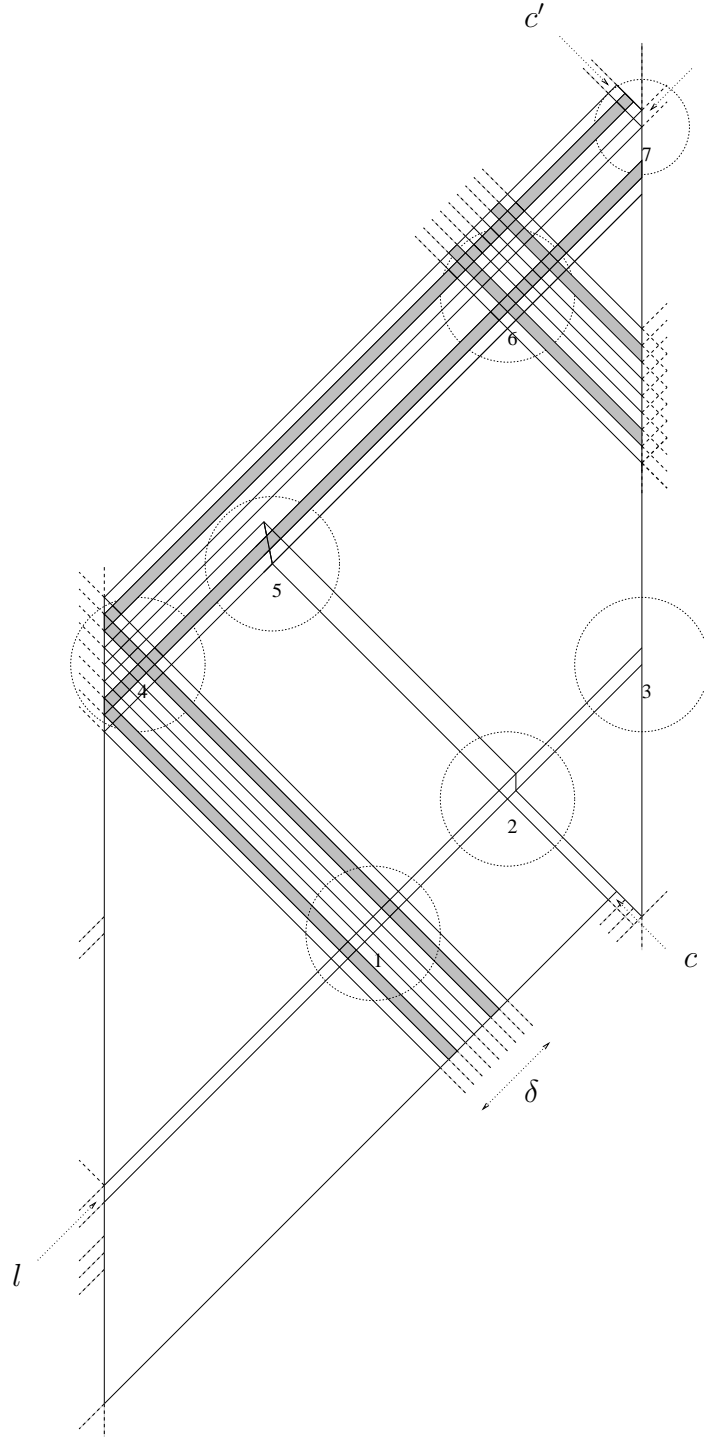
**Fig. 4.** Symbolic representation of the widget  
Plain and evenly dashed lines represent signals,  
dotted lines indicates signals that propagate to other cells

The result is a signal  $l + c$  going left and the signal  $l$  going right. This outgoing  $l$  signal will be absorbed by the vertical particle (area 3).

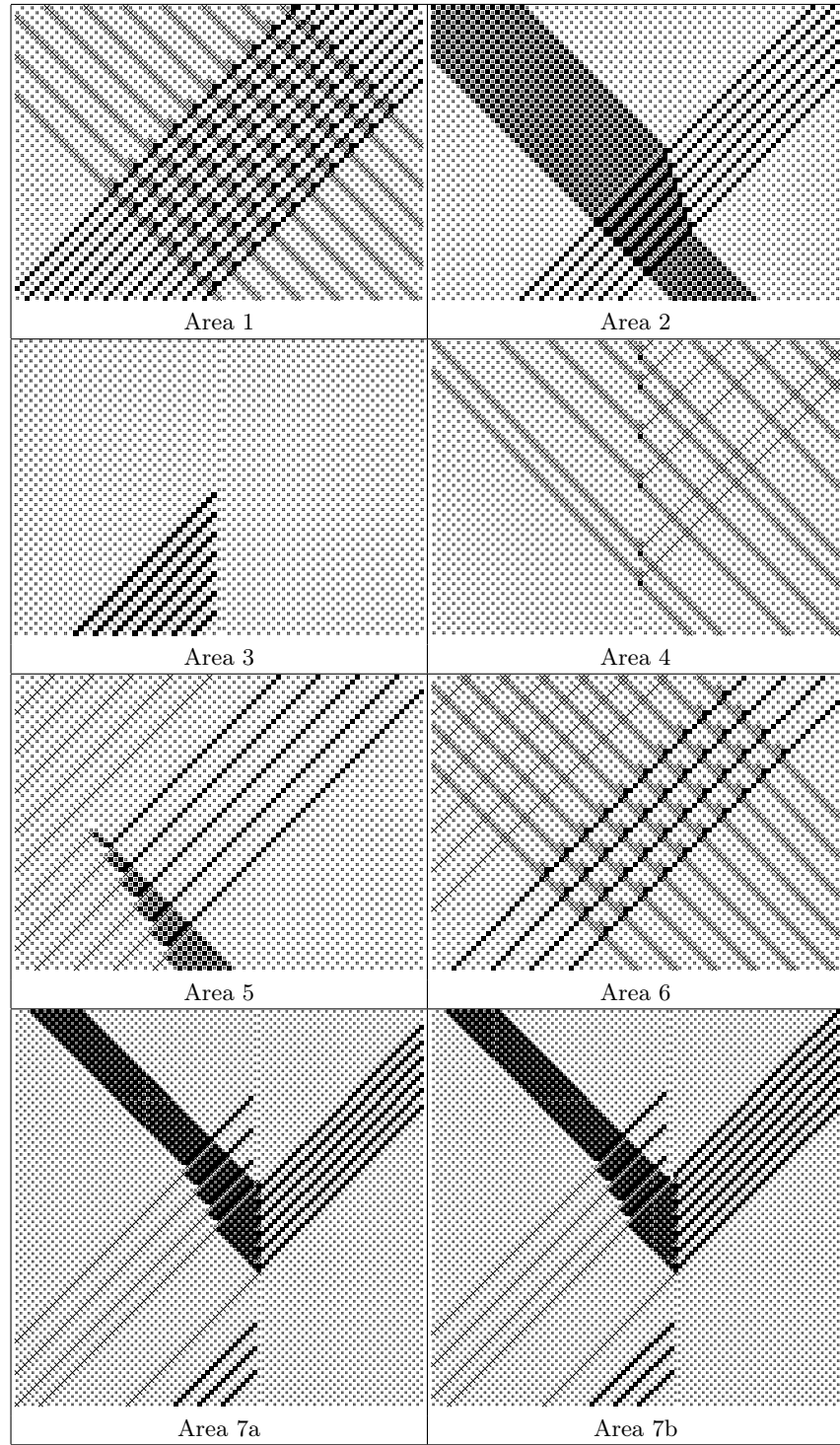
The signal  $l + c$  encounter next a copy of the function  $\delta$  (originated from collision between *delta* and the vertical particle detailed in area 4). In this encounter, the signal  $l + c$  disappear and copy of  $\delta$  has its  $l + c - 1$  first particles altered and the following one suppressed. This step allow to select the  $l + c + 1^{\text{th}}$  value of the rule.

At last, the altered copy of  $\delta$  encounter the vertical particle (area 7a and area 7b) . First altered signals are absorbed. Then the first gap (of size  $\delta_{l+c}$ ) is converted into two signals. Here, the behavior varies according to the parity of  $\delta_{l+c}$ . if  $\delta_{l+c} = 2k$  (area 7a) then  $c' = 2k - N + l + c - 1$  and  $l' = k$ ; if  $\delta_{l+c} = 2k + 1$  (area 7b) then  $c' = 2k - N + l + c - 1$  and  $l' = k + 1$ .

Note that the presence of  $l$  and  $c$  in  $c'$  will force us to have a supplementary encoding step. Before going to this final part, one must say some final arguments to prove that those widgets can be used: first main thing is the problem to ensure the synchronisation of particles. In the general case, two identical particles can lead to several distinct collisions. Here, most of the particles are as short as the



**Fig. 5.** Implementation of the widget using the automaton particles  
circle area indicate area whose implementation is given in next figure



**Fig. 6.** Details from selected area of the widget  
 These figures provides space-time diagram extracts, states are represented as follow:  
 $L$  is white,  $R$  dotted, 1 is a cross and 2 black.

background, that implies there is only one collision possible. There are however two exceptions: One is the vertical particle in areas 7 and we have treated the two cases (a and b). The other is the one in area 1. One can check that this collision also works when  $l$  is even. Second thing is to check that parallels lines in symbolic representation stay parallel. Once again, whereas it can be difficult in the general case, here we can see that either particles are not deviated by collisions, either they all undergo the same deviation (for example, all particles of the signal  $\delta$  are deviated once by the vertical particle in area 4). It is also the case of the vertical particle: the right one undergo a deviation in area 7a and 7b whereas the left one undergo this deviation on the input of  $l$ .

Using this widget as tile, it is possible for every chosen  $\delta$  to construct space-time diagrams provided some assumption are verified:  $l, c > 0$ ,  $l + c < N$  (area 5) and  $\forall i, \delta_i > 1$  (area 4). In the last part, we will show how to encode any one-way cellular automaton with hose widget.

### 2.3 Simulation and coding

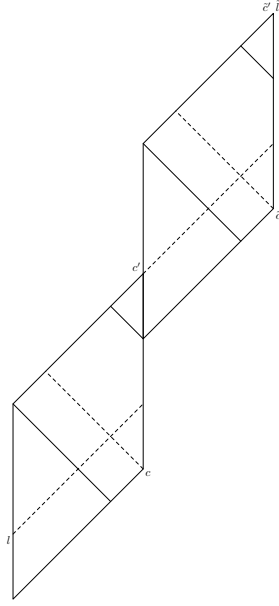
The last part of the construction will be to encode any one-way cellular automaton using the widget just built. In this widget, we can control the rule  $\delta$  according to inputs, however the outputs have strange relations with inputs. First thing will be to get rid of this.

let  $(S, f)$  be an uni-directional CA ( $f : S^2 \rightarrow S$ ). To simulate we will create a structure, based on the widget, that will emulate any cell of the original cellular automaton. The construction will be based on the assembly of two widgets: the output  $l'$  of the first one will be linked to the input  $\tilde{l}$  of the second one. The rule  $\delta$  must be the same for both. This structure takes four inputs  $l, c, \tilde{c}$  and  $\delta$ . It outputs four values  $c', \tilde{c}', \tilde{l}'$  and  $\delta$  (see fig. 7).

Assume that  $S$  is a subset of  $\{0, 1\}^k$ . let  $s^l = (s_i^l)_{0 \leq i < k}$  and  $s^c$  two states and  $s' = f(s^l, s^c)$ . This transition is simulated inside the elementary structure using coding given in table 3. In the table,  $\perp$  denotes an arbitrary value and  $\bar{x}$  is the flapped value of  $x$  (i.e.  $1 - x$ ). For transitions, we will only use the even case, that is:  $\delta(l + c) = 2k$  which lead to  $c' = 2k + l + c - (N + 1)$  and  $l' = k$ .

$N$	1111 ( 11 11 11 11 ) 111
$l$	0101 ( $\perp\perp$ $s_i^l 0$ $\perp\perp$ 00 ) $\perp\perp\perp$
$c$	0000 ( $\perp\perp$ 00 $\perp\perp$ $s_i^c 0$ ) $\perp\perp\perp$
$\tilde{c}$	1000 ( 00 00 00 00 ) 000
$l + c$	01xx ( $\perp\perp$ $s_i^l \perp$ $\perp\perp$ $s_i^c \perp$ ) $\perp\perp\perp$ where $xx = 01$ or $10$
$\delta(l + c)$	10 $\triangle\triangle$ ( $\bar{s}_i^l 1$ $\triangle\triangle$ 11 $\triangle\triangle$ ) 100
$c'$	0000 ( $\perp\perp$ 00 $\perp\perp$ $s_i^l 0$ ) $\perp\perp\perp$
$l'$	010 $\perp$ ( $\perp\bar{s}_i^l$ 1 $\perp$ $\perp 1$ 1 $\perp$ ) $\perp 10$
$N - l' + 1$	101 $\perp$ ( $\perp s_i^l$ 0 $\perp$ $\perp 0$ 0 $\perp$ ) $\perp 10$
$\tilde{l}'$	0101 ( $\perp\perp$ $s_i^l 0$ $\perp\perp$ 00 ) $\perp\perp\perp$

**Table 3.** Inner coding of a transition



**Fig. 7.** Elementary widget

The coding works the following way: first thing to note is that  $l + c$  is one-to-one with regards to  $(s^l, s^c)$ . The  $\Delta$  in  $\delta(l + c)$  can be chosen to impose the value of  $c'$ . After this, one can remark that  $l' + \tilde{c}$  is also one-to-one. If we take  $\delta(l' + \tilde{c}) = N - l' + 1$  then  $\tilde{c}' = N - l' + 1 - N + l' + \tilde{c} - 1 = \tilde{c}$ . Moreover,  $\tilde{l}'$  is on the desired form. To finish with this coding, one must say that it is possible that every states has at least one non null digit (to satisfy the assumption the  $c$  is not null).

At this point, we have shown that we can construct with an injective-surjective morphism a space-time diagram that simulates any one directional automaton. however, in this construction, configurations simulated are not horizontal but on a fixed slope (note that the slope does not depend on the simulated automaton). To overcome this, it is sufficient to use the standard trick of taking an automaton which waits enough time before really acting.

This conclude the whole proof: we have given an intrinsically universal automaton with only 4 states and show how it can simulate any other

## References

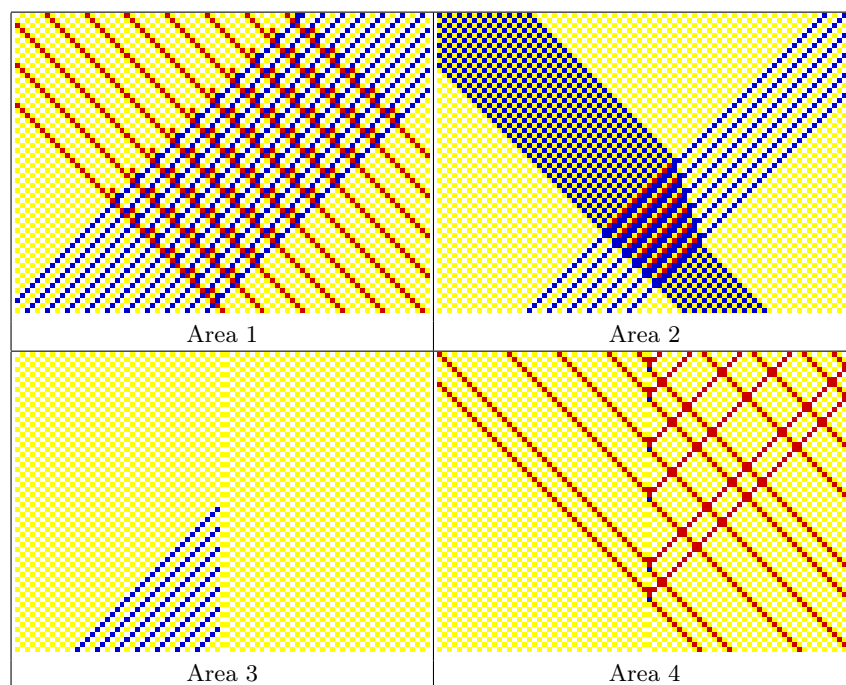
1. E.R. Banks. Universality in cellular automata. In *Symposium on Switching and Automata Theory (Santa Monica, California, 1970)*, pages 194–215. IEEE, 1970.
2. M. Cook. Universality in elementary cellular automata. *Complex Systems*, 15:1–40, 2004.

3. Bruno Durand and Zsuzsanna Róka. The game of life: universality revisited. In *Cellular automata (Saissac, 1996)*, pages 51–74. Kluwer Acad. Publ., Dordrecht, 1999.
4. J. Kari. Theory of cellular automata: A survey. *Theoretical Computer Science*, 334:3–33, 2005.
5. K. Lindgren and M.G. Nordahl. Universal computation in simple one-dimensional cellular automata. *Complex Systems*, 4(3):299–318, 1990.
6. T. Neary and D. Woods. P-completeness of cellular automaton rule. In *ICALP*, pages 132–143, 2006.
7. J. Von Neumann. *Theory of Self-Reproducing Automata*. University of Illinois Press, Urbana, Ill., 1966.
8. N. Ollinger. *Automates cellulaires : structures*. PhD thesis, École Normale Supérieure De Lyon, 2002.
9. N. Ollinger. The quest for small universal cellular automata. *Lecture Notes in Computer Science*, pages 318–329, 2002.
10. N. Ollinger. The intrinsic universality problem of one-dimensional cellular automata. In *STACS*, pages 632–641, 2003.
11. S. Wolfram. *A new kind of science*. Wolfram Media Inc., Champaign, Illinois, US, United States, 2002.

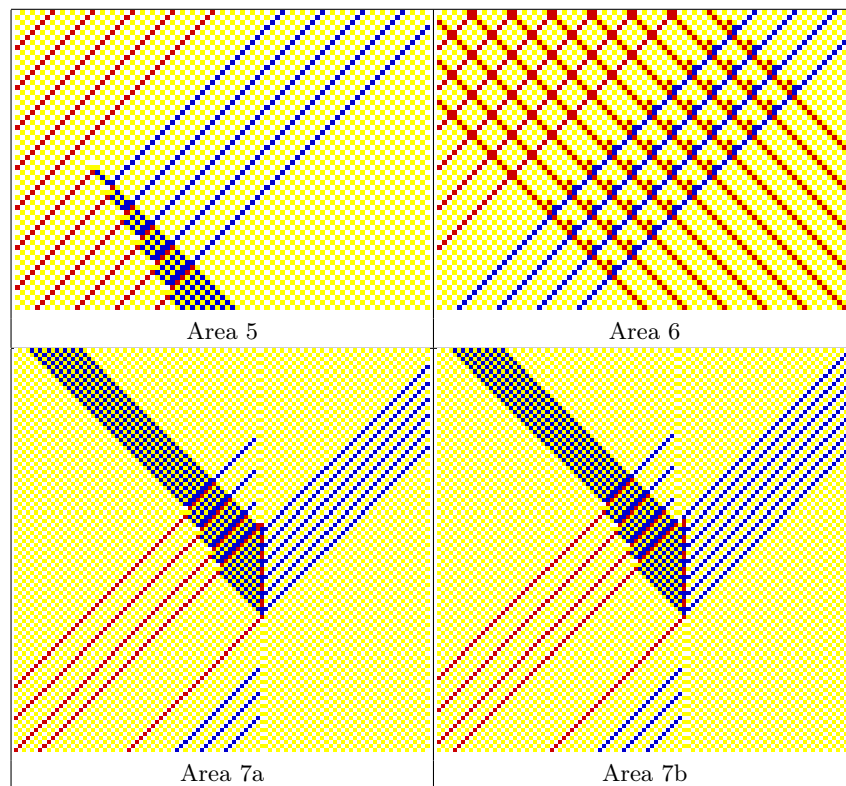
## A Colored space-time diagram

This section contains a colored version of space-time diagram extracts given in fig. 6. The color mapping is the following:  $L$  is white,  $R$  yellow, 1 red and 2 blue.





**Fig. 8.** Details in colored version from selected area of the widget



**Fig. 9.** Details in colored version from selected area of the widget (cont.)