A particular universal cellular automaton

Nicolas Ollinger, Gaétan Richard

A Particular Universal Cellular Automaton *

Nicolas Ollinger and Gaétan Richard **

Laboratoire d'informatique fondamentale de Marseille (LIF),
Aix-marseille Université, CNRS,
39 rue Joliot-Curie, 13 013 Marseille, France.

Abstract. Signals are a classical tool of cellular automata constructions that proved to be useful for language recognition or firing-squad synchronisation. Particles and collisions formalize this idea one step further, describing regular nets of colliding signals. In the present paper, we investigate the use of particles and collisions for constructions involving an infinite number of interacting particles. We obtain a high-level construction for a new smallest intrinsically universal cellular automaton with 4 states.

Introduction

Cellular automata were introduced by J. von Neumann [11] in the forties to study self-reproduction. They consist of a parallel computation model in discrete time based on an infinite grid of regular *cells* (endowed with a *state* chosen among a finite alphabet) interacting synchronously with each other. Even if it is well known that such systems have a great computational power, one outstanding question is to find the smallest universal cellular automaton. In fact, there exist two kind of universality: Turing universality consists in "simulating" any Turing machine whereas intrinsic universality consists in simulating any other cellular automaton.

For a detailled survey of universality of cellular automata, see [8]. For dimension two or above, the intrinsically universal cellular automaton constructed by E. R. Banks [1] (2 states) is optimal. For dimension one, the result of M. Cook [2] (2 states) is optimal for the case of Turing universality. However, this cellular automaton does not achieve the stronger notion of intrinsic universality, leaving a gap between Turing universality and intrinsic universality since the best result on intrinsic universality is with 6 states [6].

All the preciously mentioned constructions use meta-structures known as particles and collisions. Those elements are the core of algorithmic on cellular automaton and were heavily studied by J. Mazoyer and V. Terrier [5] . For constructions, a formal approach has been proposed in [9]. With this method, it was possible to obtain [10] a new high-level proof of the result of M. Cook. In this paper, we use this formalism to construct quite simply an intrinsically universal

* Work suppoted by a grant of the French ANR
** corresponding author: Gaetan.Richard@lif.univ-mrs.fr

cellular automaton with 4 states narrowing the gap between Turing universality and intrinsic universality. The rule of the automaton is constructed to ensure the presence of particles and collisions with a predictable behavior. The technical part of the proof is to encode information with those elements to simulate any cellular automaton. Thus this proof is quite alike complexity NP-completeness proofs.

In section 1, we first recall definitions of cellular automata and different types of universality and then introduce the formalism used for particles and collisions. In section 2, we construct the rule of the cellular automaton and show how particles and collisions can be used. In section 3, we discuss in more details the encoding used and finish the proof.

1 Cellular automata, universality and self-organisation

In this paper, we only consider cellular automata in dimension one, with the three nearest neighbours. Therefore, a *cellular automaton* is a pair (S, f) where S is a finite set of *states* and $f : S^3 \to S$ is the *local transition function*. The automaton act on elements $c \in S^{\mathbb{Z}}$ (called *configurations*) by $F(c)_i = f(c_{i-1}, c_i, c_{i+1})$. Moreover, a cellular automaton is said to be *one-way* if the local function does not depend on its third argument. A *space-time diagram* D is a bi-infinite sequence of configurations obtained in the dynamics of cellular automaton. More formally, it can be seen as an element of $S^{\mathbb{Z}^2}$ satisfying $\forall i, j \in \mathbb{Z}, D(i, j+1) = f\left(D(i-1, j), D(i, j), D(i+1, j)\right)$.

Unless many other known computation systems, cellular automata act on infinite configurations and do not posses any halting property. These two properties have led to several different definitions of universality. One position is to simulate Turing machine by using "ad-hoc" properties. General scheme of different suggested properties was summarised by B. Durand and Zs. Róka [3]. *Turing universality* requires that, for each Turing machine, there exist a computable function that transforms each input word of the Turing machine into a ultimately periodic initial configuration and another computable function to decide halt such that the initial configuration eventually reachs a configuration with halting property if and only if the Turing machine eventually halts on the input word.

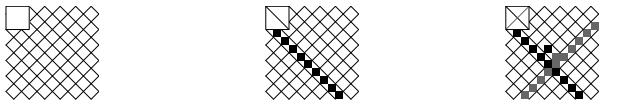
The other approach is to simulate any other cellular automata evolution. Intuitively, a cellular automaton simulates another one is the space-time diagram of the latter can be "embedded" into the former. Formally, a cellular automaton $\mathfrak{A} = (S, f)$ *simulates* a cellular automaton $\mathfrak{B} = (T, g)$ if there exists $l, m, n \in \mathbb{N}$ and $t : S^l \to S$ partial onto[1] application such that $\forall c \in T^{\mathbb{Z}}, F^m(t^{-1}(d)) \subseteq t^{-1}(G^n(d))$. That is, up to encoding by t, F^m contains the dynamic of G^n. With this notion of simulation, it is possible to define intrinsic universality.

[1] Usually, t^{-1} is requested to be one-one. Here, we choose to use this lighter version for convenience. It is an open (and very interesting) question to know whether those two definitions are equivalent.

Definition 1. *A cellular automaton is said to be* intrinsically universal *if it can simulate any other cellular automaton.*

The notion of intrinsic universality is stronger than the notion of Turing universality in the sense that every intrinsic universal cellular automata is Turing universal but the converse is false. Therefore, it is not surprising that the intrinsic universality problem (i.e., given a cellular automaton $\mathfrak{A}$, decide whether $\mathfrak{A}$ is intrinsically universal) is undecidable [7]. Throughtout the paper, universality always refers to intrinsic universality.

As many others constructions, we heavily rely on particles and collisions to encode and compute information. To define these objects, we use the approach developed in [9] seeing space-time diagrams as tilings of $S^{\mathbb{Z}^2}$ with local constraints. Therefore, we need to introduce some concepts from discrete geometry: a *coloring* $\mathcal{C}$ is an application from a subset $Sup(\mathcal{C}) \subset \mathbb{Z}^2$ to S. Such coloring is said *finite* if $Sup(\mathcal{C})$ is finite. Three natural operations on colorings are *translation* of a coloring $\mathcal{C}$ by a vector $u \in \mathbb{Z}^2$ defined by $(u \cdot \mathcal{C})(z+u) = \mathcal{C}(z)$; *disjoint union* of two colorings $\mathcal{C}$ and $\mathcal{C}'$ with $\mathrm{Sup}(\mathcal{C}) \cap \mathrm{Sup}(\mathcal{C}') = \emptyset$ defined by $\mathcal{C} \oplus \mathcal{C}'(z) = \mathcal{C}(z)(\text{resp. } \mathcal{C}'(z))$ for all $z \in \mathrm{Sup}(\mathcal{C}')(\text{resp. } \mathrm{Sup}(\mathcal{C}'))$; at last, *restriction* of a coloring $\mathcal{C}$ to $D \subset \mathbb{Z}^2$ is denoted by $\mathcal{C}_{|D}$. With those elements, let us give definitions of the three used structures of self-organisation.



(a) A background (b) A particle (c) A collision

Fig. 1. Examples of structures present in self-organisation

First structures are two-dimensional *backgrounds* (see Fig.1a) which are triplets $\mathfrak{B} = (\mathcal{C}, u, v)$ where $\mathcal{C}$ is a finite coloring and u, v two non-collinear vectors ensuring that $\bigoplus_{i,j\in\mathbb{Z}^2}(iu+jv)\cdot\mathcal{C}$ is a space-time diagram (this space-time diagram is often also referred as $\mathfrak{B}$ when no confusion is possible). These structures mostly serve to "fill" empty space in constructions. Among backgrounds, one-dimensional particles travel (see Fig.1b). *Particles* are quadruplets $\mathfrak{P} = (\mathcal{C}, u, \mathfrak{B}_l, \mathfrak{B}_r)$ where $\mathcal{C}$ is a finite coloring, u a vector, $\mathfrak{B}_l$ and $\mathfrak{B}_r$ two backgrounds ensuring that $\mathcal{I} = \bigoplus_{k\in\mathbb{Z}} ku\cdot\mathcal{C}$ separates the plane in two 4-connected domains L and R (L being the left-one according to u) so that $\mathfrak{B}_{|L} \oplus \mathcal{I} \oplus \mathfrak{B}'_{|R}$ is a space-time diagram. Particles are often used to convey part of information either alone or in groups consisting of parallel particles. Last but not least, *collisions* (see Fig.1c) are pair $(\mathcal{C}, L)$ where $\mathcal{C}$ is a finite coloring, L is a finite sequence of n particles $\mathfrak{P}_i = (\mathfrak{B}_i, \mathcal{C}_i, u_i, \mathfrak{B}'_i)$, satisfying the following conditions: first, consecutive particles on the list agree on their common background ($\forall i \in \mathbb{Z}_n, \quad \mathfrak{B}'_i = \mathfrak{B}_{i+1}$), then particles and finite perturbation form a star ($\mathcal{I} = \mathcal{C} \oplus \bigoplus_{i\in\mathbb{Z}_n, k\in\mathbb{N}} ku_i \cdot \mathcal{C}_i$ cuts the plane in n 4-connected zones and For all $i \in \mathbb{Z}_n$, $\mathcal{C} \oplus \bigoplus_{k\in\mathbb{N}} (ku_i \cdot \mathcal{C}_i \oplus ku_{i+1} \cdot \mathcal{C}_{i+1})$ cuts the plane in two 4-connected

zones. Let P_i be the one right of $\mathfrak{P}_i$) and at last, $\mathfrak{C} = \mathcal{I} \oplus \bigoplus_i \mathfrak{B}_{i\restriction P_i}$ is a space-time diagram.

To describe easily complex space-time diagrams, one idea is to symbolise particles as lines and collisions as points, giving birth to a planar map called catenation scheme as the one in upper-left corner of Fig. 3. Formally, a *catenation scheme* is a planar map whose vertices are labeled by collisions and edges by particles which are coherent with regards to collisions. Since catenation schemes are symbolic representations, it is not clear that there exist associated space-time diagrams. In fact, to go back from a catenation scheme to a "real" space-time diagram, one must give explicit relative positions of collisions as, for example, by giving the number of repetitions for each particle (edge) of the scheme. Such set of integers is called *affectation* and is said to be *valid* is the resulting object is a space-time diagram. The main result on catenation scheme is that set of valid affectations can be computed from finite catenation schemes.

Theorem 1 ([9]). *Given a finite catenation scheme, the set of valid affectations is a computable semi-linear set.*

In this paper, the constructed automaton is based on particles and collisions and thus, heavily relies on the methodology used for catenation scheme. However, since the particles and collisions are explicitly constructed, they are very small and posses many good combinatorial properties. Therefore we do not need the whole power of catenation scheme and can often give simpler arguments for our specific case. The rest of the paper is devoted to construct the automaton and prove it is intrinsically universal. This is done in two steps: first, we explicit the automaton and show, using catenation scheme, that it can somehow "simulate" the local behavior of any automaton. Then, we show how to assemble those local simulations into a global one.

2 The automaton and elementary block

In this section, we present the automaton and show how it can "somehow" simulate any local transition function of one-way cellular automaton in an *elementary block*. Before going on with this block, one point to notice is that it is sufficient to simulate one-way cellular automata to be intrinsically universal. Therefore, our elementary block is constructed such that it takes three inputs (one transition function and two arguments) and outputs three values (one transition function and two "copies" of the result).

To provide a good view of the automaton, all needed materials are depicted in Fig. 2 and Fig. 3. The first figure gives explicit local transition function and particles, whereas the second one gives the catenation scheme along with all interesting extracts of corresponding space-time diagram. The rest of this section is devoted to describe and explain the contents of these two figures. The local transition function is fully depicted on the top of Fig. 2. In fact, it is not a real transition rule since some cases (denoted by white) are not used and thus can take arbitrary values.

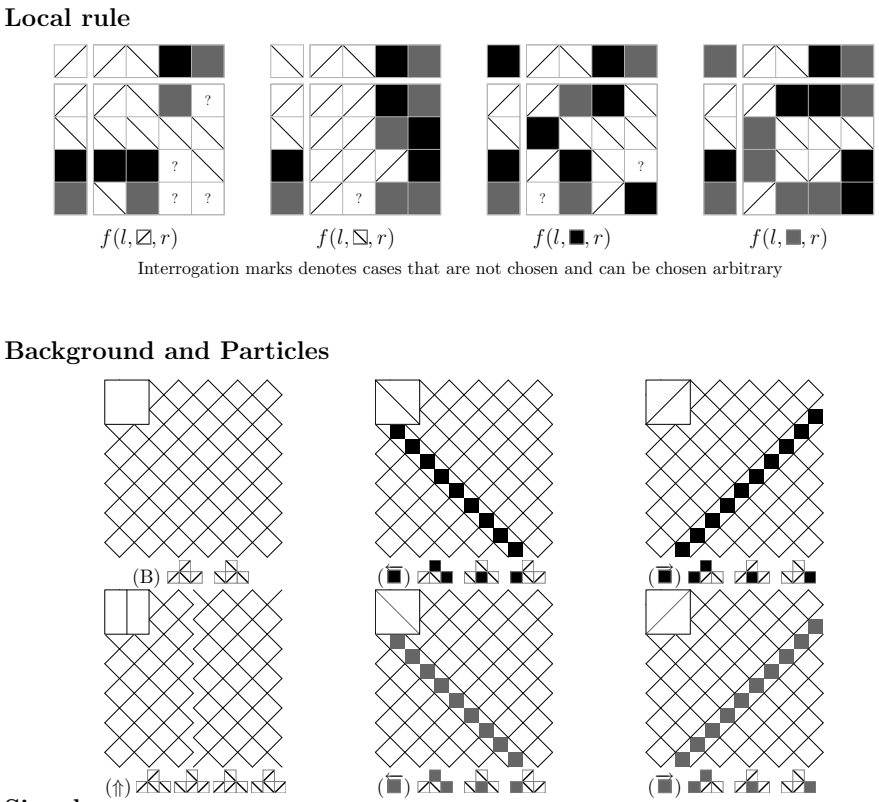


Fig. 2. Local rule and particles

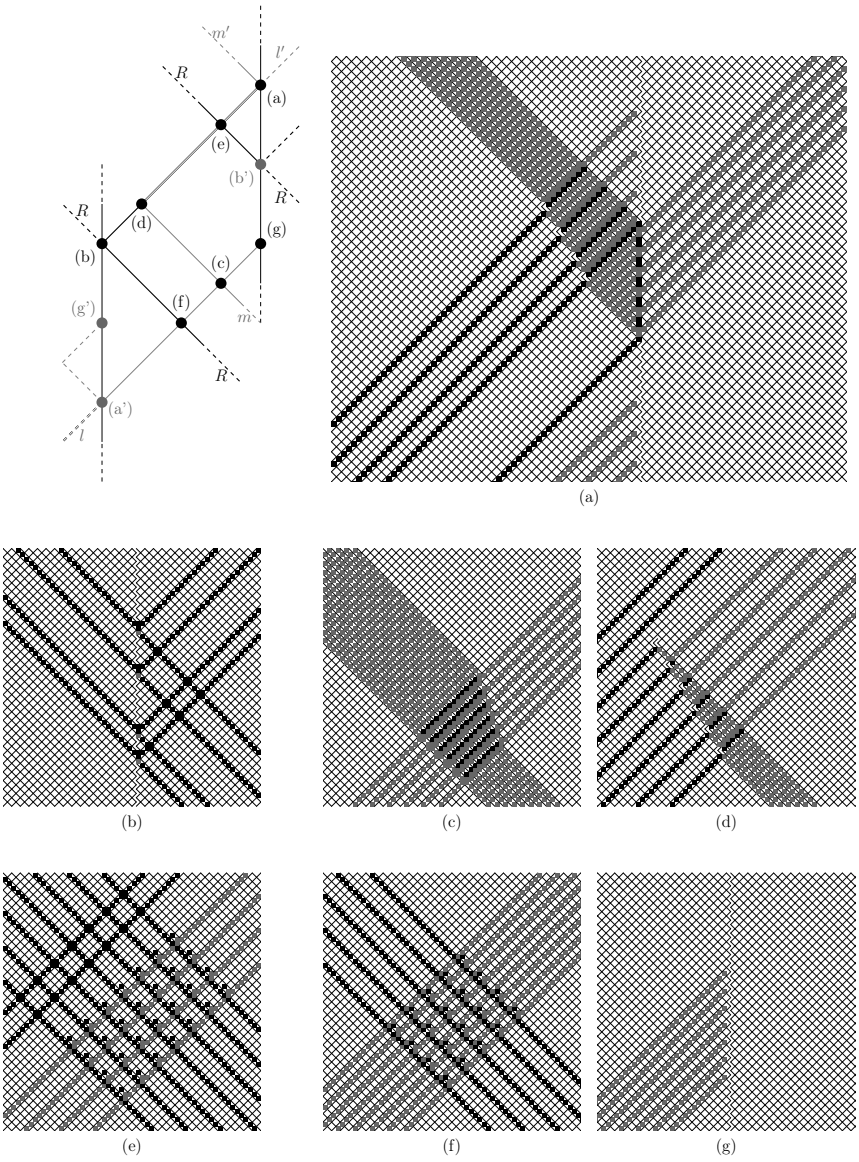


Fig. 3. Constructed block and collisions

Even if giving the rule could be sufficient, the next part is devoted to give intuitions behind local transition function to ease understanding of the automaton. The first part is devoted to present background and particles used in the construction. Those structures are depicted in the corresponding part of Fig. 2. For each structure, we give a meaningful extract of space time, its name along with used local transition function cases. The formal definition can be trivially extracted from those diagrams.

One main specificity of our cellular automaton is that, contrary to other known constructions, it does not use a uniform background but a bi-colored check-board (B). With this new approach, we need two states (instead of one) for background but this allows us to have a greater range of particles since each remaining state can lead to two distinct particles according to its alignment within the background. Thus, with the two remaining states, we can construct four different particles ($\overleftarrow{\blacksquare}$, $\overrightarrow{\blacksquare}$, $\overleftarrow{\blacksquare}$, $\overrightarrow{\blacksquare}$). Furthermore, since background has two different phases, one can construct an additional particle ($\Uparrow$) by taking advantage of the gap between the two phases. Those are the main particles used in the construction.

To encode information, the basic idea is to use groups of parallel particles that we call *signals*, depicted in the corresponding section of Fig. 2. In those group, information can be encoded in two different ways: either by the number and type of the particles used or by relative position between those particles. Here, we use both approaches. As noted before, we need to encode three main information: the *left state* (i.e., state of the left neighbour) is encoded in unary by the number of (regularly spaced) $\overrightarrow{\blacksquare}$ particles (signal l). The same way, the *center state* is encoded in unary by the number of (regularly spaced) $\overleftarrow{\blacksquare}$ particles (signal m). For the *local transition function*, it is seen as an array of integers and encoded by spaces between particles $\overleftarrow{\blacksquare}$ (signal R): the j-th element of the array corresponds to the space between the j-th and the $j+1$-th particle of the signal. To be exact, the space is counted by the sum of numbers of ◪ between two consecutive ■ and two (for the previously mentioned states). This way of counting may seem a little obscure but is chosen to give nice formulae in the end of this section. In the rest of the paper, names of symbols are also used when speaking of encoded values.

During computation, other kind of signals are appearing: First, a mirror copy of signal R (R') which encode the same information using $\overrightarrow{\blacksquare}$ particles (going in the opposite direction). At last, an altered version $\tilde{R}'$ (of R') appear. In this altered version, some of the leftmost $\overrightarrow{\blacksquare}$ particles are replaced by $\overrightarrow{\blacksquare}$. Due to choice of background and particles, our automaton has a special property which makes the proof a lot easier. Contrary to many other cellular automata, we have no synchronisation problems.

Lemma 1. *When two of the particles (or signals) described above collide, occurring collision is always the same.*

Proof. As studied by J. P Crutchfield *et al.* [4], the number of ways two particles (or signals) can collide depends on the relative position between those two par-

ticles. Possible relative positions takes into account repetition vectors of those particles and constraints induced by backgrounds. In our case, repetition vectors are either $(1,1)$, $(0,2)$ or $(-1,1)$ which suggests the possibility of two distinct collisions. However, the background forbids one of those possibility leaving only one possible case. □

With this very strong property, we can focus on symbolic behavior. Let us go and construct the dynamic. The scheme of elementary block is depicted in Fig. 3 (along with extracts of all induced collisions). The block is made the following way: left and right borders are delimited by $\Uparrow$ particle. At the bottom, we have a left state signal l, a rule R and a center state signal m. The left state signal is going through the rule (collision f) and then collides with center state signal (collision c). This collision outputs an unused copy of the left signal to the right. This signal is erased when encountering the right border (collision g). Collision c is also sending a signal encoding the the sum of left and center state to the left. This new signal is encountering (collision d) the mirror copy R' (created when R crossed the left border during collision b). During collision d, as many particle of R' are altered[2] than encoded value (i.e., the sum of left and center signals). The signal embedding the sum is destroyed by the collision whereas the altered rule $\tilde{R}$ proceed to the right. After crossing another rule R (collision e), the altered rule $\tilde{R}$ collides with the right border and produces at the same time a new center state signal and a left state signal (located right of the border) in collision a.

Additional cases needed for each collision in the local transition function are made explicit at the bottom of Fig. 3. This scheme gives us a symbolic block which somehow "computes" the rule. Now, let us prove that this symbolic behavior does really correspond to a valid space-time diagram and look at the details at the computed function. The following proposition deals with the first problem by ensuring that, under reasonable conditions, the scheme of symbolic block is a valid space-time diagram. Moreover, it gives some additional results on regularity of this block.

Proposition 1. *For any encoded rule R which contains only even number greater that four, for any reasonable encoded value in left and right state signal (i.e. they are not null and their sum is less that the number of integers encoded in R) the scheme of the symbolic block correspond to a valid space-time diagram. Moreover, the size of the space time diagram does not depend on the encoded states and those blocks can tile the plane.*

Proof. First of all, previous lemma ensures us that collisions occurring have all the same behavior; non null condition ensures that all particles exist. The proof that collisions are valid can be directly deduced from extracts given in Fig. 3. Due to the fact that constraints are local, periodicity considerations are sufficient to prove the validity of collisions occurring. The case of collisions c, e, f and g

[2] Last alteration is an erasing rather than a change of the particle but this does not change the behavior

is easily get rid of since the perturbation inside the collision is periodic: adding one (or more) particles is the same as increasing the size of the periodic portion of the collision. Collision b requires space between particles $\overleftarrow{\blacksquare}$ to be greater than four. For collision d, the constraint is just that there are at least one particles $\overrightarrow{\blacksquare}$ left (i.e. sum of left and right values is less that the number of integers encoded in R). The last significant point is in collision a: since the vertical portion in the collision has periodicity $(0, 4)$, it requires that inter-space in $\tilde{R}$ being even (this inter-space is of course the same as in R). If all these constraints are respected (which is the case for our proposition) then the resulting catenation scheme is a valid space-time diagram.

Now let us fix R, this implies that encoded values in states signals are bounded. Therefore, all signals are of bounded size and thus can be rendered "nearly punctual" by increasing the size of the cell. With this, it is sufficient to take relative positions of signal as in the scheme of Fig. 3 to achieve same size blocks.

The last point is to prove that such space-time diagrams, associated to elementary blocks, tile the plane. First remark is that symbolic blocks already tile the plane. Now if we look in details, border are left untouched by all collisions and rule signals are only shifted of a constant when encountering a border. This implies that crossings of rule signals and border (collisions b) form a regular grid on the plane. At last, position of all other collisions only depends on a small number of neighbour points on this grid. It is directly visible for e and a. Positions of collisions f, g and c also depend on a' but its position is fixed by previous case. At last, collision d depends on c which was already treated. □

With this result, we have an elementary block able to do simple computation according to a rule R and which can tile the plane. Before combining those elementary blocks in the next section, we must first study exactly which function is computed by our block. For the same periodicity reason than previously, it is sufficient to look at what happens in the case of collision a in Fig. 3.

Lemma 2. *Let $R(i)_{1\leq i\leq N}$ be the value encoded in R and l (resp. m) be the one encoded in l (resp. m); then the block leaves R unchanged and outputs m' and l' with encoded values respectively $R(l+c)+l+c-(N+1)$ and $R(l+c)/2$.* □

This block computation is alike a cell in a cellular automaton except that it sends different values for left and right states. This difference prevents us to give a direct simulation and require quite additional work to get rid of this problem. A better way to overcome this problem would be to alter the automaton or use unused cases in the local transition function to ensure equality of outputs. However, for the moment, we do not manage to do such a thing. Therefore we present an alternative (and quite combinatorial) solution to use this elementary blocks in intrinsic simulation in the next section.

3 Simulation of cellular automata

To achieve intrinsic universality with our cellular automaton, the idea is to use the constructed elementary blocks and encode information not directly but only on portions of integers. First step is to remark that rather than working on only one block, we can work on chain of two consecutive blocks chained by left output (see Fig. 4). The new constructed element has three inputs (namely l,m, $\tilde{m}$) and three outputs l', m' and $\tilde{m}'$ (not including rule). The key points is that we see input as integers written in binary and work on bits of these integers. The basic idea is to simulate a cell of a cellular automaton using l and m has inputs, l' and m' has outputs and maintaining the same constant value in all $\tilde{m}$ and $\tilde{m}'$.

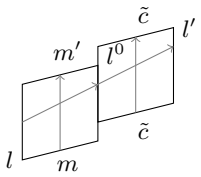


Fig. 4. Symbolic chaining of elementary blocks (encoding of state if ommited)

Let us take a one-way cellular automaton (S, f). Since f does not depends on its third arguments, we can see it has an elements of $S^2 \to S$. Every element of S can be seen as a non null integer and written as a finite word of fixed size k on binary alphabet denoted by $s = s_0 s_1 \dots s_{k-1}$. All values encoded in signals can be seen as integers with $4 + 8k + 3$ bits (including leading zeros). The first four one are called *header*, the last three one are called *footer* and each inside block of 8 (called *digit*) is used to encode one bit of the state. The size of the rule N is chosen as $2^{4+8k+3} - 1$. Only some portions of those encoding do support information. Other portions contain garbage which prevents to achieve one-one encoding.

The method used to encode and decode is depicted in Tab. 1. The encoding of l and m contains some junk but is chosen so that $l + m$ contains the whole information about states encoded in l and m. With this, it is possible to chose the value of $R(l + m)$ according to these states. Half of the bits of $R(l + m)$ are chosen to ensure the value of l^0. The other half (denoted by symbol $\bullet$) is used to ensure correct values in $m' = R(l + m) + l + m - (N + 1)$. In the second cell, one observe $l^0 + \tilde{m}$ which has a different header and thus a different value of any valid $l + m$. Therefore, one can chose $R(l^0 + m)$ as $N - l^0 + 1$ (note that this value is even). With this choice, $\tilde{m}' = N - l^0 + 1 + l^0 + \tilde{m} - N - 1 = \tilde{m}$ and l' is on the correct form. With this encoding, we can simulate any rule of cellular automaton inside our cells. Leading to the following theorem:

Theorem 2. *The cellular automaton presented in Fig 2 is intrinsically universal.*

	header	(	digits				)	footer	
N	1111	(	11	11	11	11	)	111	
l	0101	(	$\perp\perp$	s_i0	$\perp\perp$	00	)	$\perp\perp\perp$	
m	0000	(	$\perp\perp$	00	$\perp\perp$	s'_i0	)	$\perp\perp\perp$	
$\tilde{c}$	1000	(	00	00	00	00	)	000	
$l+m$	$01xx$	(	$\perp\perp$	$s_i\perp$	$\perp\perp$	$s'_i\perp$	)	$\perp\perp\perp$	where $xx = 01$ or 10
$R(l+m)$	$10\bullet\bullet$	(	$\bar{t}_i1$	$\bullet\bullet$	11	$\bullet\bullet$	)	100	
m'	0000	(	$\perp\perp$	00	$\perp\perp$	t_i0	)	$\perp\perp\perp$	
l^0	$010\perp$	(	$\perp\bar{t}_i$	$1\perp$	$\perp1$	$1\perp$	)	$\perp10$	
$N-l^0+1$	$101\perp$	(	$\perp t_i$	$0\perp$	$\perp0$	$0\perp$	)	$\perp10$	
l'	0101	(	$\perp\perp$	t_i0	$\perp\perp$	00	)	$\perp\perp\perp$	

$\perp$ denotes arbitrary value and $\bullet$ value to be fixed.

Table 1. Encoding of states inside values

Proof. With the previously presented encoding, chains of elementary blocks can encode any one-way cellular automaton local transition rule. Moreover, the encoding ensures conditions for proposition 1. Thus constructed elementary blocks can tile the plane. If we associate to each cell of the simulated cellular automaton space-time diagram the corresponding (up to garbage consideration) chain of elementary blocks, we can simulate any other automaton. Last point is to note that even if the simulation is not horizontal, the problem can be easily overcome with standard tricks. □

Conclusion

Our construction heavily relies on signals and manages to give a construction with two disctint levels: local rule is constructed to ensure a set of particles and collisons. Then, the simulation is made encoding the computation with this set. This method allows us to have a clear and understandable construction which separates the local and global aspect and gives us a new smallest intrisically universal cellular automaton.

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. B. Durand and Zs. Róka. The game of life: universality revisited. In *Cellular automata (Saissac, 1996)*, pages 51–74. Kluwer Acad. Publ., Dordrecht, 1999.
4. W. Hordijk, C. R. Shalizi, and J. P. Crutchfield. Upper bound on the products of particle interactions in cellular automata. *Phys. D*, 154(3-4):240–258, 2001.
5. J. Mazoyer and V. Terrier. Signals in one-dimensional cellular automata. *Theoretical Computer Science*, 217(1):53–80, 1999. Cellular automata (Milan, 1996).

6. N. Ollinger. The quest for small universal cellular automata. In *ICALP '02: Proceedings of the 29th International Colloquium on Automata, Languages and Programming*, pages 318–329, London, UK, 2002. Springer-Verlag.
7. N. Ollinger. The intrinsic universality problem of one-dimensional cellular automata. In *STACS '03: Proceedings of the 20th Annual Symposium on Theoretical Aspects of Computer Science*, pages 632–641, London, UK, 2003. Springer-Verlag.
8. N. Ollinger. Universalities in cellular automata; a (short) survey. In *JAC 2008*, 2008. (to appear).
9. N. Ollinger and G. Richard. Collisions and their catenations: Ultimately periodic tilings of the plane. In *IFIP-TCS*, 2008. (to appear) http://hal.archives-ouvertes.fr/hal-00175397/.
10. G. Richard. Rule 110: universality and catenations. In *JAC 2008*, 2008. (to appear).
11. J. Von Neumann. *Theory of Self-Reproducing Automata.* University of Illinois Press, Urbana, Ill., 1966.