Communication Complexity and Intrinsic Universality in Cellular Automata

Eric Goles Chacc, Pierre-Etienne Meunier, Ivan Rapaport, Guillaume Theyssier

Communication Complexity
and Intrinsic Universality
in Cellular Automata[*]

E. Goles[a], P.-E. Meunier[c], I. Rapaport[b], G. Theyssier[c,*]

[a]*Facultad de Ingenieria y Ciencias, Universidad Adolfo Ibáñez, Santiago, Chile*
[b]*DIM, CMM (UMI 2807 CNRS), Universidad de Chile, Santiago, Chile*
[c]*LAMA, Université de Savoie, CNRS, 73 376 Le Bourget-du-Lac Cedex, France*

Abstract

The notions of universality and completeness are central in the theories of computation and computational complexity. However, proving lower bounds and necessary conditions remains hard in most of the cases. In this article, we introduce necessary conditions for a cellular automaton to be "universal", according to a precise notion of simulation, related both to the dynamics of cellular automata and to their computational power. This notion of simulation relies on simple operations of space-time rescaling and it is intrinsic to the model of cellular automata. *Intrinsic universality*, the derived notion, is stronger than Turing universality, but more uniform, and easier to define and study.

Our approach builds upon the notion of *communication complexity*, which was primarily designed to study parallel programs, and thus is, as we show in this article, particulary well suited to the study of cellular automata: it allowed us to show, by studying natural problems on the dynamics of cellular automata, that several classes of cellular automata, as well as many natural (*elementary*) examples, were not *intrinsically universal*.

Keywords: cellular automata, communication complexity, intrinsic universality.

1. Introduction

Since the pioneering work of J. von Neumman [15], universality in cellular automata (CA) has received a lot of attention (see [12] for a survey). Historically, the notion of universality used for CA was more or less an adaptation of the classical Turing-universality. Later, a stronger notion called *intrinsic universality* was proposed: A CA is intrinsically universal if it is able to simulate any other CA [3, 9, 12] through a uniform and regular encoding based on *rescaling*.

This definition of intrinsic universality may seem very restrictive. However, it can be very common among natural families of CA [1], and allows a complete

---

and precise formalization of the notion of universality [2]. As we are going to see, this preciseness, and the robustness of this definition, allows for concrete proofs of negative results and lower bounds.

Indeed, in this paper we will explain how to rule out particular elementary cellular automata, as well as whole well-known classes of cellular automata, from being intrinsically universal, using the elegant framework of communication complexity.

In Section 2 we give the basic definitions. One of the key definitions is the following: Given a traditional computational problem $\mathcal{P}$ with an arbitrary input w, we can split the input into two subwords w_1 and w_2; therefore, we can refer to the "comunication complexity" of such problem (w_1 is given to Alice while w_2 is given to Bob).

In Section 3 we introduce a family of "canonical problems" concerning various aspects of the dynamics of a given CA. In other words, for any CA F and any prototype problem $\mathcal{P}$, we consider the problem $\mathcal{P}_F$.

In Section 4 we explain how to infer properties of F from the study of the communication complexity of $\mathcal{P}_F$. More precisely, we prove that if the communication complexity of one of our canonical problem $\mathcal{P}_F$ is not maximal, then F is not intrinsically universal. In other words, we are introducing a powerful tool for ruling out CA from being intrinsically universal. We conclude that linear, expansive and reversible CA are not intrinsically universal. We also show the uncomparability of our three canonical problems: none of them is sufficient to discard *all* non-universal cellular automata, and none of them is stronger than any other.

In Section 5 we explain clearly why the communication complexity approach appears to be a promising tool for ruling out CA from being intrinsically universal. More precisely, we prove computational intractability results about problems that our framework considers very simple.

Finally, in Section 6 we use our results to prove that a few concrete elementary CA are not intrinsically universal. Although looking at several space-time diagrams of these automata might give a strong intuition about their non-universality, we stress that producing complete formal proofs for such a negative result is a difficult task and, as far as we know, had never been done before.

2. Basic definitions

2.1. Communication complexity

Communication complexity is a notion introduced by A. C.-C. Yao in [16], and designed at first for lower-bounding the amount of communication needed in distributed algorithms. In that model he considered two players, namely Alice and Bob, both with arbitrary computational power and communicating to each other in order to collaboratively decide the value of a given function. More precisely, for a function $\phi : X \times Y \to Z$, the question is "how much information do they need to exchange, in the worst case, in order to compute $\phi(x, y)$, with Alice knowing only x and Bob only y".

---

[2] There is actually no consensus on the formal definition of Turing-universality in CA (see [3] for a discussion about encoding/decoding problems).

This communication problem is solved by a *protocol*, which specifies, at each step of the communication between Alice and Bob, who speaks (Alice or Bob), and what she/he says (a bit, 0 or 1), as a function of her/his respective input. This simple framework, and some of its variants we discuss in this article, appears to be promising for studying CA.

A protocol $\mathcal{P}$ over a domain $X \times Y$ with range Z is a binary tree where each internal node v is labeled either by a map $a_v : X \to \{0, 1\}$ or by a map $b_v : Y \to \{0, 1\}$, and each leaf v is labeled either by a map $A_v : X \to Z$ or by a map $B_v : Y \to Z$.

The *value* of protocol $\mathcal{P}$ on input $(x, y) \in X \times Y$ is given by $A_v(x)$ (or $B_v(y)$) where A_v (or B_v) is the label of the leaf reached by walking on the tree from the root, turning left if $a_v(x) = 0$ (or $b_v(y) = 0$), and right otherwise. We say that a protocol computes a function $\phi : X \times Y \to Z$ if for any $(x, y) \in X \times Y$, its value on input (x, y) is $\phi(x, y)$.

Intuitively, each internal node specifies a bit to be communicated either by Alice or by Bob, whereas at the leaves either Alice or Bob determines the value of ϕ when she/he has received enough information from the other party.

In our formalism, we do not ask both Alice and Bob to be able to give the final value. We do so in order to consider protocols where communication is unidirectional.

We denote by $\mathbf{cc}(\phi)$ the (deterministic) communication complexity of a function $\phi : X \times Y \to Z$. It is the minimal cost of a protocol, over all protocols computing ϕ, where the cost of a protocol is the depth of its corresponding tree.

One approach for proving lower bounds on the communication complexity of an arbitrary function ϕ is based on the so-called *fooling sets* (for a deeper presentation of this theory we refer to [8]).

Definition 1. *Given a function $\phi : X \times Y \to Z$, a set $S \subseteq X \times Y$ is a* fooling set *for ϕ if there exists $z \in Z$ with:*

1. *$\forall (x, y) \in S, \phi(x, y) = z$,*
2. *$\forall (x_1, y_1) \in S, \forall (x_2, y_2) \in S$, either $\phi(x_1, y_2) \neq z$ or $\phi(x_2, y_1) \neq z$.*

The usefulness of fooling sets is given by the following lemma (see [8]).

Lemma 1. *If S is a fooling set of size m for ϕ then $\mathbf{cc}(\phi) \geq \log_2(m)$.*

In addition to ad hoc fooling set constructions, we will use the following classical lower bounds on communication complexity (the proofs appear in [8]).

Proposition 1. *Let $n \geq 1$ be fixed. Let ϕ_{EQ}, ϕ_{IP} and ϕ_{DISJ} be the functions "equality", "inner product" and "disjointness" defined from $\{0, 1\}^n \times \{0, 1\}^n$ to $\{0, 1\}$ by:*

$$\phi_{\mathrm{EQ}}(x, y) = \begin{cases} 1 & \textit{if } (\forall i)(x_i = y_i), \\ 0 & \textit{otherwise.} \end{cases}$$

$$\phi_{\mathrm{IP}}(x, y) = \begin{cases} 1 & \textit{if } \sum_i x_i y_i \bmod 2 = 1, \\ 0 & \textit{otherwise.} \end{cases}$$

$$\phi_{\mathrm{DISJ}}(x, y) = \begin{cases} 1 & \textit{if } (\forall i)(x_i y_i \neq 1), \\ 0 & \textit{otherwise.} \end{cases}$$

The following lower bounds hold:

- $\mathbf{cc}(\phi_{\mathrm{EQ}}) \geq n$.

- $\mathbf{cc}(\phi_{\mathrm{IP}}) \geq n$.

- $\mathbf{cc}(\phi_{\mathrm{DISJ}}) \geq n$.

2.2. Splitting the input of computational problems

Let us consider now classical computational input-output problems. In this work we will only encounter problems of the form $\mathcal{P} : Q^* \to Z$, whose inputs are words over some alphabet Q and outputs are elements of a finite set Z. Moreover, we will always have $Z = Q$ or $Z = \{0,1\}$ as output sets.

Given such type of problem $\mathcal{P}$, we define, for any n, its restriction to words of length n; i.e, we consider the restricted problem $\mathcal{P}|_n : Q^n \to Z$.

The key idea of the communication approach is to *split* the input into two parts: For any $1 \leq i \leq (n-1)$, we define $\mathcal{P}|_n^i : Q^i \times Q^{n-i} \to Z$. More precisely, for every $x \in Q^i, y \in Q^{n-i}$, we have $\mathcal{P}|_n^i(x,y) = \mathcal{P}|_n(xy)$. Then, we can consider the communication complexity $\mathbf{cc}(\mathcal{P}|_n^i)$ of the ith split function $\mathcal{P}|_n^i$. Of course the choice of i matters and can alter the corresponding communication complexity. Since we don't want to rely on an arbitrary choice, we consider the worst case. This yields the following definition:

Definition 2. *Let $\mathcal{P} : Q^* \to Z$ be a problem. The* communication complexity *of $\mathcal{P}$, denoted* $\mathrm{CC}(\mathcal{P})$, *is the function:*

$$n \mapsto \max_{1 \leq i \leq n-1} \mathbf{cc}(\mathcal{P}|_n^i).$$

2.3. Cellular automata

In this paper we are always going to consider one-dimensional CA. A CA is defined by its local rule $f : Q^{2r+1} \to Q$ (where Q corresponds to the set of states and r denotes the radius of the local rule). For any $n \geq 2r+1$, we extend $f : Q^{2r+1} \to Q$ to the more general $f : Q^n \to Q^{n-2r}$ by

$$f(u_1 \cdots u_n) = f(u_1 \cdots u_{2r+1}) \cdots f(u_{n-2r} \cdots u_n).$$

Moreover, for every $1 \leq t \leq \lfloor (n-1)/2r \rfloor$, we define the t-steps local iteration as $f^t : Q^n \to Q^{n-2 \cdot r \cdot t}$ by

$$\begin{cases} f^1 = f \\ f^t(u_1 \cdots u_n) = f\big(f^{t-1}(u_1 \cdots u_{n-2r}) \cdots f^{t-1}(u_{2r+1} \cdots u_n)\big) \end{cases}$$

We also define $f^* : Q^* \to Q^*$ by

$$f^*(u) = f^{\left\lfloor \frac{|u|-1}{2r} \right\rfloor}(u).$$

Intuitively, f^* applied on u consist in iterating f as long as possible (until ending up with a word too short for f). The result is a word of length at most $2r$ (depending on $|u| \bmod 2r$).

We denote by $F : Q^{\mathbb{Z}} \to Q^{\mathbb{Z}}$ the global rule induced by f following the classical definition:

$$F(c)_z = f(c_{z-r}, \ldots, c_{z+r}).$$

Finally, we denote by $F^t : Q^{\mathbb{Z}} \to Q^{\mathbb{Z}}$ the t-step iteration of the global function F.

A global function F can be represented by different local functions. All properties considered in this paper depend only on F and are not sensitive to the choice of a particular local function. However, to avoid useless formalism, we will use the following notion of *canonical* local representation: (f, r) is the canonical local representation of F if f has radius r and it is the local function of smallest radius having F as its associated global function.

Throughout this work we are going to refer to the CA F with (f, r) being its canonical local representation.

3. The three canonical communication problems

In this section we define the three "problem schemes" on which we are going to apply the communication complexity approach. Before entering into details, we stress that this set of problems tackles various dynamical aspects of CA: Transient, periodic and asymptotic regime starting respectively from finite, cyclic, or ultimately periodic configurations. Moreover, algorithmically speaking, they are also very different since they belong respectively to the classes P, PSPACE, and Π_1^0 (and can be complete for these classes as we will see in this section).

Thus, they form an interesting set of prototype problems.

3.1. Prediction

The prediction problem consists in determining the far future of a cell given the state of sufficiently many cells around it.

Definition 3. *Let F be a CA. The problem $\mathrm{PRED}_F : Q^* \to Q$ is defined as follows:*

$$\mathrm{PRED}_F(u) = \big(f^*(u)\big)_1,$$

where (f, r) is the canonical local representation of F while the "$\big(f^*(u)\big)_1$" notation means that we take the first letter of the word $f^*(u)$, which has length at most $2r$.

Clearly, this problem is in $\mathrm{DTIME}(n^2)$, and, as we have already said before, we can also view PRED_F as a communication problem (see Figure 1): Given an initial configuration as input, we *split* the initial configuration between Alice and Bob, and ask for the *final* value computed by F on this input configuration, as represented in Figure 1(b).

More precisely, for every $1 \le i \le (n-1)$, $\mathrm{PRED}_F|_n^i : Q^i \times Q^{n-i} \to Q$ is such that $\mathrm{PRED}_F|_n^i(x, y) = (f^*(xy))_1$. This function $\mathrm{PRED}_F|_n^i$ can be represented as a $|Q|^i \times |Q|^{n-i}$ matrix. In other words, we give i states to Alice (rows) and $n - i$ states to Bob (columns); i.e. $X = Q^i$ and $Y = Q^{n-i}$. We denote by $M_F^{n,i}$ such a matrix. In the examples of Figure 2, we have $n = 2i + 1 = 13$ and $n = 2i + 1 = 15$ (for the elementary CA Rule 178).

Remark. *We can consider the more restricted one-round communication complexity. In this setting only one party (either Alice or Bob) is allowed to send information. This restriction is justified by the fact that, according to a theorem of [8], by simply counting the number of different rows or columns of a certain*
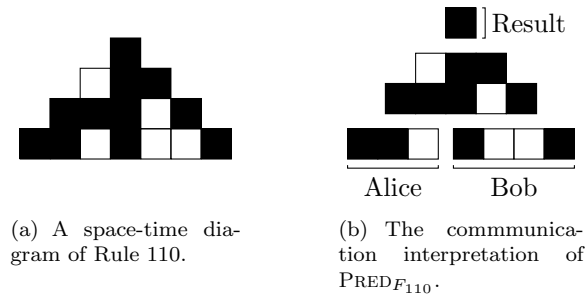
(a) A space-time diagram of Rule 110.

(b) The commmunication interpretation of $\textsc{Pred}_{F_{110}}$.

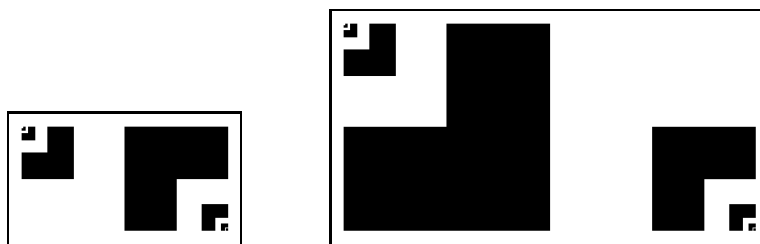Figure 1: Problem $\textsc{Pred}_{F_{110}}$.

Figure 2: Matrices $M_{F_{178}}^{13,6}$ and $M_{F_{178}}^{15,7}$, where "178" stands for the elementary CA Rule 178.

matrix we obtain the *exact one-round communication complexity of the function*. In our framework, the one round communication complexity of $\textsc{Pred}_F|_n^i$ corresponds to the minimum between the number of different rows and different columns of $M_F^{n,i}$. Therefore, performing computational experiments in order to infer the one-round communication complexity of $\textsc{Pred}_F|_n^i$, becomes an easy task.

Recall that, given a CA F, the communication complexity of $\textsc{Pred}_F$ is defined as:

$$\mathrm{CC}\left(\textsc{Pred}_F\right) = n \mapsto \max_{1 \leq i \leq n-1} \mathbf{cc}\left(\textsc{Pred}_F|_n^i\right).$$

Remark. *In the above definition of* $\textsc{Pred}_F$*, we choose a canonical local representation* (f, r) *for the CA* F*. Replacing* f *by another valid local representation can change the problem and its communication complexity. However this change would only introduce a multiplicative factor and therefore would not alter the main point of this paper (Section 4.3).*

Now we show that some well-known properties of CA induce small upper bounds for the communication complexity of the prediction problem. The results below are adaptations of ideas of [4] to the formalism adopted in the present paper.

Proposition 2. *Let* F *be any CA and* (f, r) *be its canonical local representation. If there is a function* $g : \mathbb{N} \to \mathbb{N}$ *such that* f^n *depends on only* $g(n)$ *cells, then* $\mathrm{CC}\left(\textsc{Pred}_F\right) \leq g(n)/2$.

In the work of M. Sablik [13], CA which are equicontinuous in some direction are considered. Following Theorem 4.3 of [13], they have a bounded number of dependant cells (i.e, a bounded function $g(n)$). A well known example of such CA are the nilpotent CA (a CA is *nilpotent* if it converges to a unique configuration from any initial configuration, or equivalently, if F^t is a constant function for any large enough t).

Corollary 1. *If F is an equicontinuous CA in some direction then*

$$\text{CC}\,(\text{PRED}_F) \in O(1).$$

Another set of CA with that property is the set of linear CA. A CA F with state set S is linear if there is an operator $\oplus$ such that $(S, \oplus)$ is a semi-group with neutral element e and for all configurations c and c' we have:

$$F(c \,\overline{\oplus}\, c') = F(c) \,\overline{\oplus}\, F(c'),$$

where $\overline{\oplus}$ denotes the uniform (cell-by-cell) extension of $\oplus$.

Proposition 3. *If F is a linear CA then* $\text{CC}\,(\text{PRED}_F) \in O(1)$

Proof. The proof appears in [4] in a different setting. The idea is that there is a simple one-round protocol to compute linear functions: Alice and Bob can each compute on their own the image the function would produce assuming the other party has only the neutral element as input; then Alice or Bob communicate this result to the other who can answer the final result by linearity. $\square$

3.2. Invasion

Let F be a CA and let u be a given word. Roughly, the problem INV_F^u is defined as follows: Given an input word w, we define the u-periodic configuration p_u on the one hand, and the configuration $p_u(w)$ obtained by putting the word w at the origin over p_u on the other hand; the invasion problem consists in determining whether the differences between p_u and $p_u(w)$ will expand to an infinite width as time tends to infinity (hatched surface on Figure 3).

As we show in Proposition 5.2, the general case is, from the point of view of classical algorithmic theory, undecidable.
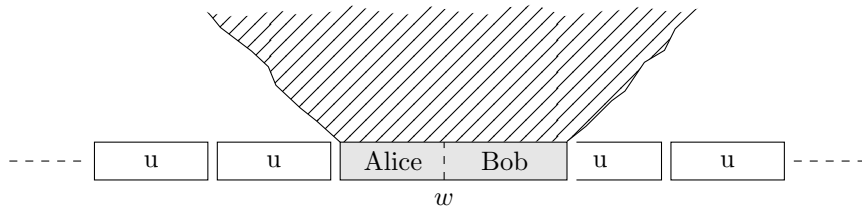

Figure 3: The INVASION problem

Now we give formal definitions.

Definition 4. *Let $u = u_1 \ldots u_l$ be a finite word. Let p_u be such that for all $i \in \mathbb{Z}$, $p_u[i] = u[i \mod l]$.*

- *we consider the ultimately periodic orbit $\left(F^t(p_u)\right)_t$ as the reference orbit;*

- *for each $x_1, \ldots, x_n \in Q$, we define the configuration $p_u(x_1, \ldots, x_n)$ obtained by modifying p_u as follows:*

$$p_u(x_1, \ldots, x_n)_z = \begin{cases} (p_u)_z & \text{for } z \leq 0 \text{ or } z \geq n+1, \\ x_z & \text{otherwise.} \end{cases}$$

- *for each t, we denote $\delta_l(t)$ and $\delta_r(t)$ the lefmost and rightmost differences between the tth images of p_u and $p_u(x_1, \ldots, x_n)$:*

$$\delta_l(t) = \min\left\{z : F^t(p_u)_z \neq F^t\left(p_u(x_1, \ldots, x_n)\right)_z\right\},$$
$$\delta_r(t) = \max\left\{z : F^t(p_u)_z \neq F^t\left(p_u(x_1, \ldots, x_n)\right)_z\right\}.$$

- *then $\text{INV}_F^u(x_1 \ldots x_n)$ equals 1 if $\delta_r(t) - \delta_l(t) \to_t \infty$ and 0 otherwise.*

As explained before, we associate to any F and u, the communication complexity of INV_F^u defined as $\text{CC}\left(\text{INV}_F^u\right)$.

Some CA have by nature a trivial invasion complexity because their dynamics consists in propagating errors systematically. This is the case of (positively) expansive CA. Recall that F is (positively) expansive if there is some $\epsilon > 0$ such that:
$$\forall x, y, \ x \neq y \Rightarrow \exists t, d(F^t(x), F^t(y)) \geq \epsilon$$

where d is the Cantor distance.

Proposition 4. *Let F be a positively expansive CA. Then for all u we have $\text{CC}\left(\text{INV}_F^u\right) = 1$.*

Proof. Fix any u and consider any $(x_1, \ldots, x_n)$ such that $p_u(x_1, \ldots, x_n) \neq p_u$. By classical results of P. Kůrka [7], there is a positive constant α (average propagation speed) such that $\delta_l(t) \leq -\alpha t$ and $\delta_r(t) \geq \alpha t$. Therefore, invasion occurs if and only if:
$$p_u(x_1, \ldots, x_n) \neq p_u.$$

Testing this condition can be done with only 1 bit of communication: Either Alice or Bob communicates whether she (or he) sees any difference between her (or his) input and the corresponding part of p_u; then the other party can answer. The proposition follows. $\qquad\square$

3.3. Cycle length

For this last problem, we consider spatially periodic configurations. Since there are only a finite number of such configurations of a given period size, and the size of the period does not grow with time, then clearly the evolution becomes periodic (in time) after a certain number of steps (see Figure 4 where successive steps are represented by successive concentric circles). Roughly speaking, the CYCLE problem consists in determining whether the length of this ultimate (temporal) period is small, starting from a given (spatially) periodic initial configuration. The formal definition follows.
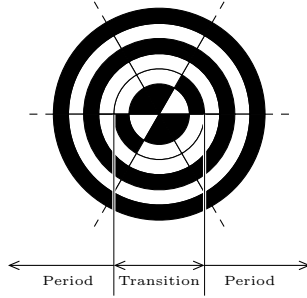
Figure 4: The CYCLE problem on elementary CA Rule 33. Since the configurations are cyclic, we can represent one configuration on a circle. Time goes from the inner circle to outer circles, zeros are white, and ones are black. For instance, the initial configuration – on the innermost circle – is 011011. After one step, it becomes 100100.

Definition 5. *Let F be a CA and let $k \geq 1$. For any $u \in Q^*$ we denote by $\lambda(u)$ the length of the ultimate period of the orbit of configuration p_u under F:*

$$\lambda(u) = \min\big\{p : \exists t_0, \forall t \geq t_0, F^t(p_u) = F^{t+p}(p_u)\big\}.$$

The problem CYCLE_F^k is then defined by:

$$\text{CYCLE}_F^k(u) = \begin{cases} 1 & \text{if } \lambda(u) \leq k, \\ 0 & \text{otherwise.} \end{cases}$$

One of the interests of the cycle length problem lies in the following complexity upper bound for reversible CA.

Proposition 5. *Let F be any reversible CA. Then, for any fixed k, we have:*

$$\text{CC}\left(\text{CYCLE}_F^k\right) \in O(1).$$

Proof. For a reversible CA, orbits of periodic configurations are not only ultimately periodic but also periodic. More precisely, for any periodic configuration c, the cycle length starting from c is less than k if and only if:

$$\exists t \leq k : F^t(c) = c.$$

Thus, Alice and Bob can simply simulate the automaton for k steps, then check if a configuration repeats during these steps : this can be done with $4k \cdot r \cdot \lfloor 1 + \log Q \rfloor$ bits, to transmit the cells next to the border between Alice and Bob's respective parts, then one bit for Alice to tell Bob if a configuration appeared twice during the k steps.
$\qquad\square$

4. The three corresponding necessary conditions for intrinsic universality

In this section we show that intrinsic universality implies that the communication complexity of the three canonical problems described above must be

9

maximal. Before giving precise definitions, recall that a CA is intrinsically universal if it is able to simulate any other CA. Our approach with communication complexity proceeds in two steps:

- we show that the simulation of F by G implies a reduction from any canonical problem for F to the corresponding problem for G in such a way that the communication complexity is preserved (up to some distortions involving only multiplicative factors);

- we show the existence of maximal communication complexity CA for each of the canonical problems.

Before developing these two steps, we give formal definitions for simulations and intrinsic universality.

4.1. Simulations and universality

The base ingredient is the relation of sub-automaton. A CA F is a *sub-automaton* of a CA G, denoted by $F \sqsubseteq G$, if there is an injective map ι from Q_F to Q_G such that $\bar{\iota} \circ F = G \circ \bar{\iota}$, where $\bar{\iota} : Q_F^{\mathbb{Z}} \to Q_G^{\mathbb{Z}}$ denotes the uniform extension of ι.

A CA F simulates a CA G if some *rescaling* of G is a sub-automaton of some *rescaling* of F. The ingredients of the rescalings are simple: packing cells into blocks, iterating the rule and composing with a translation. Formally, given any state set Q and any $m \geq 1$, we define the bijective packing map $b_m : Q^{\mathbb{Z}} \to \left(Q^m \right)^{\mathbb{Z}}$ by:

$$\forall z \in \mathbb{Z} : \left(b_m(c) \right)(z) = \left(c(mz), \ldots, c(mz + m - 1) \right)$$

for all $c \in Q^{\mathbb{Z}}$. The rescaling $F^{<m,t,z>}$ of F by parameters m (packing), $t \geq 1$ (iterating) and $z \in \mathbb{Z}$ (shifting) is the CA of state set Q^m and global rule:

$$b_m \circ \sigma_z \circ F^t \circ b_m^{-1}.$$

The fact that the above function is the global rule of a cellular automaton follows from Curtis-Lyndon-Hedlund theorem [6] because it is continuous and commutes with translations. With these definitions, we say that G simulates F, denoted $F \preccurlyeq G$, if there are rescaling parameters m_1, m_2, t_1, t_2, z_1 and z_2 such that $F^{<m_1,t_1,z_1>} \sqsubseteq G^{<m_2,t_2,z_2>}$.

We can now naturally define the notion of universality associated to this simulation relation.

Definition 6. *F is* intrinsically universal *if for all G it holds that $G \preccurlyeq F$. F is* reversible universal *if for all reversible G it holds that $G \preccurlyeq F$.*

We consider the following relation of comparison between functions from $\mathbb{N}$ to $\mathbb{N}$:

$$\phi_1 \prec \phi_2 \iff \exists \alpha, \beta, \gamma, \delta \geq 1, \forall n \in \mathbb{N} : \phi_1(\alpha n) \leq \beta \phi_2(\gamma n) + \delta.$$

Remark. *All the functions we will compare by $\prec$ are in $O(n)$ since they come from a communication complexity problem. Moreover, the set of such functions that are in $\Omega(n)$ form an equivalence class for $\prec$. Although we sometimes give more precise bounds, most of the paper focuses on whether or not some function belongs to this class.*

Proposition 6. *If $F \preccurlyeq G$ then $\mathrm{CC}\left(\mathrm{PRED}_F\right) \prec \mathrm{CC}\left(\mathrm{PRED}_G\right)$.*

Proof. We successively consider each "ingredient" involved in the simulation relation.

Sub-automaton: if $F \sqsubseteq G$ then each valid protocol to compute $\mathrm{PRED}_G|_n^i$ is also a valid protocol to compute iterations of $\mathrm{PRED}_F|_n^i$ (up to state renaming).

Iterating: We have $\mathrm{CC}\left(\mathrm{PRED}_{F^t}\right) \in \Theta(\mathrm{CC}\left(\mathrm{PRED}_F\right))$. In fact, if we have a protocol for the prediction problem of F^t – which is an automaton of radius $t \cdot r$ – then we can use it to predict F : on a configuration x of size n, we use the protocol to predict the result of iterating $\lfloor\frac{n}{r\cdot t}\rfloor$ times F^t, which gives a configuration of size at most $t \cdot r - 1$. To do this, we just use the protocol at most $t \cdot r - 1$ times to predict each cell of this configuration, then Alice or Bob conclude by simulating the automaton directly.

The other direction is even simpler: a protocol for PRED_F can be used directly for PRED_{F^t} by just slightly reducing the input of Bob.

Shifting: This operation only affects the splitting of inputs. Since we always take in each case the splitting of maximum complexity, this has no influence on the final complexity function.

Packing: let F be any CA and n be fixed. Consider the problem $\mathrm{PRED}_{F^{<m,1,0>}}|_n^j$ for some j. Now consider any sequence of valid protocols (P_i), one for each problem $\mathrm{PRED}_F|_{nm}^i$. It follows from the the definition of packing maps that $\mathrm{PRED}_{F^{<m,1,0>}}|_n^j$ can be solved by applying m suitably chosen protocols in the sequence (P_i). Therefore

$$\mathrm{CC}\left(\mathrm{PRED}_{F^{<m,1,0>}}\right)(n) \leq m \cdot \mathrm{CC}\left(\mathrm{PRED}_F\right)(n)$$

Reciprocally, one has for all n:

$$\mathrm{CC}\left(\mathrm{PRED}_F\right)(n) \leq \mathrm{CC}\left(\mathrm{PRED}_{f^{<m,1,0>}}\right)(\lceil n/m \rceil) + m$$

where the additional constant m is used to deal with input splittings of $\mathrm{PRED}_F|_n$ which have no equivalent in $\mathrm{PRED}_{f^{<m,1,0>}}|_{\lceil n/m \rceil}$ because they do not cut the input at a position which is multiple of m.

Therefore we have: $\mathrm{CC}\left(\mathrm{PRED}_F\right) \prec \mathrm{PRED}_{F^{<m,t,z>}}$, $\mathrm{PRED}_{F^{<m,t,z>}} \prec \mathrm{CC}\left(\mathrm{PRED}_F\right)$ and if $F \sqsubseteq G$ then $\mathrm{CC}\left(\mathrm{PRED}_F\right) \prec \mathrm{PRED}_G$. The proposition follows. $\square$

The following result shows that the invasion complexity is increasing with respect to simulations.

Proposition 7. *If $F \preccurlyeq G$ then for all u there is v such that*

$$\mathrm{CC}\left(\mathrm{INV}_F^u\right) \prec \mathrm{CC}\left(\mathrm{INV}_G^v\right).$$

Proof. The simulation relation $\preccurlyeq$ is such that ultimately periodic configurations of F are converted into ultimately periodic configurations of G. Hence, the invasion problem of F reduces to the invasion problem of G. More precisely, it is sufficient to check the following properties, each dealing with an aspect of the simulation relation $\preccurlyeq$:

- for any CA F, any u and any rescaling parameters m, t, z, we have

$$\mathrm{CC}\left(\mathrm{INV}_F^u\right) \prec \mathrm{CC}\left(\mathrm{INV}_{F^{<m,t,z>}}^U\right)$$

where U is the period of the configuration $b_m(p_u)$;

- if $F \sqsubseteq G$ then, for any u, $\mathrm{CC}\left(\mathrm{INV}_F^u\right) \prec \mathrm{CC}\left(\mathrm{INV}_G^u\right)$;

- for any CA F, any rescaling parameters m, t, z, any U (over the alphabet of $F^{<m,t,z>}$) $\mathrm{CC}\left(\mathrm{INV}_{F^{<m,t,z>}}^U\right) \prec \mathrm{CC}\left(\mathrm{INV}_F^u\right)$ where u is the period of the configuration $b_m^{-1}(p_U)$.

The result follows by composition of the 3 properties above. $\qquad\square$

Finally, we show a similar result for the cycle length problem. The problem is parametrized by an integer k and the following proposition establishes that for suitable but arbitrary large values of this parameter the complexity of the problem is conserved.

Proposition 8. *If $F \preccurlyeq G$ then for all k_0 there is k and k' such that:*

- $k \geq k_0$ *and* $k' \geq k_0$;

- $\mathrm{CC}\left(\mathrm{CYCLE}_F^k\right) \prec \mathrm{CC}\left(\mathrm{CYCLE}_G^{k'}\right)$.

Proof. The effect of rescaling transformations on cyclic orbits of periodic configurations is to change the (spatial) period length as well as the (temporal) cycle length. More precisely, we have:

- if $F \sqsubseteq G$ then, for any k, $\mathrm{CC}\left(\mathrm{CYCLE}_F^k\right) \prec \mathrm{CC}\left(\mathrm{CYCLE}_G^k\right)$;

- for any k,

  - $\mathrm{CC}\left(\mathrm{CYCLE}_F^k\right) \prec \mathrm{CC}\left(\mathrm{CYCLE}_{F^{<m,1,0>}}^k\right)$ and

  - $\mathrm{CC}\left(\mathrm{CYCLE}_{F^{<m,1,0>}}^k\right) \prec \mathrm{CC}\left(\mathrm{CYCLE}_F^k\right)$;

- for any t and any k we have:

$$\mathrm{CC}\left(\mathrm{CYCLE}_{F^{<1,t,0>}}^k\right) \prec \mathrm{CC}\left(\mathrm{CYCLE}_F^{kt}\right);$$

- for any t and any k such that $k \bmod t = 0$ we have:

$$\mathrm{CC}\left(\mathrm{CYCLE}_F^k\right) \prec \mathrm{CC}\left(\mathrm{CYCLE}_{F^{<1,t,0>}}^{k/t}\right).$$

The proposition follows. $\qquad\square$

4.2. Existence of CA with maximal complexity

This section is devoted to the following existence result.

Proposition 9.

1. *There exists a reversible CA F and a word u with $\mathrm{CC}\left(\mathrm{INV}_F^u\right) \in \Omega(n)$.*
2. *There exists a reversible CA F with $\mathrm{CC}\left(\mathrm{PRED}_F\right) \in \Omega(n)$.*
3. *There exists a CA F s.t. for any $k \geq 1$, $\mathrm{CC}\left(\mathrm{CYCLE}_F^k\right) \in \Omega(n)$.*

We now define the reversible CA of assertion 2 of Proposition 9, which we call G in the sequel. It is made of 3 layers:

- flag layer $Q_f = \{0, 1\}$,

- circulation layer $Q_c = \{W\} \cup \{0, 1\} \times \{0, 1\}$,

- test layer $Q_t = \{0, 1\} \times \{0, 1\}$.

The flag layer is simply the identity over Q_f. The circulation layer does not depend on other layers and has the following behaviour.
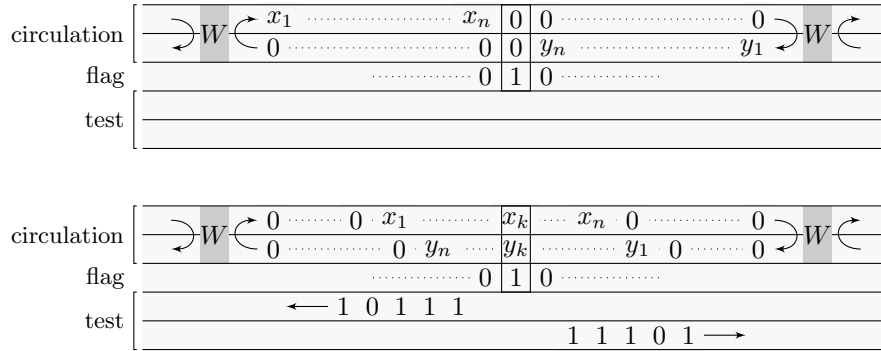

Figure 5: Above: initial configuration. Below: the configuration k steps later.

- normal states in $\{0, 1\} \times \{0, 1\}$ represent two sub-layers (top and bottom) and, if no W state is in the neighbourhood, the top sub-layer simply shifts to the right and the bottom sub-layer simply shifts to the left.

- W states are walls: They stay unchanged forever. Moreover, a normal cell on the right of a wall has the following behaviour: The top value shifts to the right and the bottom value goes to the top. A normal cell on the left of a wall has a symmetric behaviour: The bottom value shifts to the left and the top value goes to the bottom. See figure 5.

Finally, the test layer is made of two sub layers (top and bottom) wich are independant. The top layer does the following:

- if the flag layer of the cell is 1 and if the circulation layer contains the state $(1, 1)$ then invert bit and shift right;

- in any other case, simply shift right.

The bottom sub-layer does the same but replace right by left.

Proof of Proposition 9.

1. We first show that G defined above has the properties of assertion 1 of the proposition. First, it is reversible: the flag and circulation layers are themselves reversible, and the knowledge of these two layers makes the flag layer reversible too.

   Now let q_0 be the state where flag layer is 0, circulation layer is $(0,0)$ and the test layer is $(0,0)$. Consider input bits $x_1, \ldots, x_n$ on the one hand and $y_1, \ldots, y_n$ on the other hand. Let X_i be the state with flag layer 0, test layer $(0,0)$ and circulation layer $(x_i, 0)$. Similarily let Y_i be the state with flag layer 0, test layer $(0,0)$ and circulation layer $(0, y_i)$. Let M be the state of flag layer 0, circulation layer W and test layer $(0,0)$. Finally let T be the state of flag layer 1, circulation layer $(0,0)$ and test layer $(0,0)$. Consider the configuration $C(x_1, \ldots, x_n, y_1, \ldots, y_n)$:

   $$^{\omega}q_0 \ M \ X_n \cdots X_1 \ T \ Y_1 \cdots Y_n \ M \ q_0^{\omega}$$

   We can consider this configuration as an instance of the invasion problem $\text{INV}_{F^{2n+3}}^{u}$ where $u = q_0$. The only possible invasion in such an instance comes from the test layer. It follows from the definition of G that there is invasion on this instance if and only if

   $$\exists i, x_i = y_i = 1.$$

   Hence, the DISJOINTNESS problem reduces to the invasion problem through such instances. Using proposition 1, we conclude that $\text{CC}\left(\text{INV}_G^{q_0}\right) \in \Omega(n)$.

2. Assertion 2 of the proposition can be proven with a CA F simpler than G, but using similar ideas. F has radius 1 and its state set is the product of 3 components:

   - left circulation with state set $\{0, 1\}$,
   - right circulation with state set $\{0, 1\}$,
   - test with state set $\{0, 1\}$.

   The behaviour is the following:

   - each of the left and right circulation components are independent of the other components and consists in simple shift (left and right respectively),
   - the test component simply flips its value if both left and right circulation components have value 1 and stays unchanged else.

   F is clearly reversible (circulation layers are independent shifts and test layer is reversible knowing other components). Moreover, the inner product problem reduces to the prediction problem of F. Indeed, for any $x, y \in \{0, 1\}^n$ consider the word

   $$w = X_1 \cdots X_n Z Y_n \cdots Y_1$$

   where X_i is the state equal to x_i on the right circulation component and 0 elsewhere, Y_i is the state equal to y_i on the left circulation component and

14

0 else, and Z is the state equal to 0 everywhere. It follows from definition of F that

$$\text{PRED}_F|_n(w) = 1 \iff \sum x_i y_i \bmod 2 = 1.$$

proposition 1 implies that $\text{CC}(\text{PRED}_F) \in \Omega(n)$.

3. We use the problem DISJ to build a hard CYCLE problem. The idea is that if Alice and Bob receive two disjoint sets as their inputs, our CA will check DISJ forever. Otherwise it will erase all the tape, leaving a uniform, 1-periodic, configuration.

   We use three layers in this construction, let us call the corresponding rules F_1, F_2 and F_3. They are all of radius one, and all use the same set of states $\{0, 1, K\}$. The K state is used to erase all three tapes: thus, if it appears on any component, it spreads on all three.

   On (local) configurations not involving K, F_1 is a simple left shift, and F_2 a simple right shift. We use F_3 as a control layer: we need to check if the two other components represent two disjoint sets. The correponding bitwise operation is:

$$\bigwedge_{i=1}^{n} \neg(x_i \wedge y_i)$$
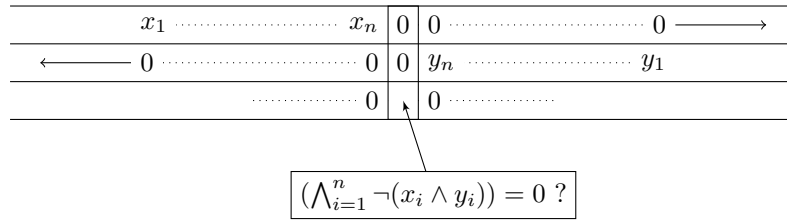

Figure 6: An automaton with a hard CYCLE problem, and an easy INV.

This corresponds to the following (partial) rule:

$$F_3\left(*, \begin{pmatrix} * \\ * \\ 0 \end{pmatrix}, *\right) = 0$$

$$F_3\left(*, \begin{pmatrix} * \\ * \\ 1 \end{pmatrix}, *\right) = 1$$

$$F_3\left(*, \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, *\right) = K$$

We consider a cyclic configuration containing an input for Alice on the first layer, and an input for Bob on the second layer, (as in Figure 6), and a third layer everywhere empty, except for a central "test" state, actually performing the tests. While the test value is 1, the tests go on. There are three cases:

- If both Alice and Bob receive the empty set, the configuration is 1-periodic, but Alice and Bob can detect this case with a single bit of communication.

- Else, since the tape is cyclic, if $\bigwedge_{i=1}^{n} \neg(x_i \wedge y_i) = 1$, then the test goes on forever, producing a (temporal) cycle of length $\Omega(n)$, because in this case, at least one x_i or one y_i is 1, and it is separated from the next 1 (possibly itself !) by at least the $2n + 1$ zeros depicted on figure 6.

- Otherwise, the test becomes 0 at some step and a spreading state is generated, which erases all the layers in both directions and produce a (temporal) cycle of length 1.

Thus, except in the case where both sets are empty, this *is* an "implementation" of the DISJ problem, shown in $\Omega(n)$ for several variants of communication complexity in [8]. This proves that this automaton can embed an $\Omega(n)$ communication problem in some of its configurations, which is enough to prove that its CYCLE problem is hard.

$\square$

Remark. *We prove in section 4.4.4 that the last construction of proposition 9 has an* INV *problem in* $O(1)$.

4.3. Necessary conditions for universality

The following corollary is the main tool provided by this paper to prove negative results about (intrinsic) universality.

Corollary 2. *Let F be an intrinsically universal CA. Then it holds that:*

1. *there exists u s.t. $\mathrm{CC}\left(\mathrm{INV}_F^u\right) \in \Omega(n)$,*
2. *$\mathrm{CC}\left(\mathrm{PRED}_F\right) \in \Omega(n)$,*
3. *there exists k s.t. $\mathrm{CC}\left(\mathrm{CYCLE}_F^k\right) \in \Omega(n)$.*

Moreover, if F is only reversible-universal, then 2 and 1 still holds.

Proof. It follows from Propositions 6, 7 and 8 on the one hand, and Proposition 9 on the other hand. $\square$

A first application of this corollary to the complexity upper-bounds presented in Section 3 yields the following necessary conditions for universality. The first proofs of these results appears in [14]. However, our approach allows us to formulate much simpler and more elegant proofs.

Corollary 3. *Let F be an intrinsically universal CA, then F cannot be:*

- *neither expansive*

- *nor linear*

- *nor reversible.*

Moreover, a reversible universal CA can not be expansive or linear.

16

4.4. Uncomparability of the three conditions

Here we show the "orthogonality" of our three problems: For any pair of problems $(\mathcal{P}_0, \mathcal{P}_1)$, we exhibit two CA, $\mathcal{A}$ and $\mathcal{B}$, such that:

- $\mathrm{CC}\left(\mathcal{P}_0^{\mathcal{A}}\right) \in o(\mathrm{CC}\left(\mathcal{P}_1^{\mathcal{A}}\right))$, in which case we say that $\mathcal{A}$ is "hard" for $\mathcal{P}_1$ and "easy" for $\mathcal{P}_0$.

- $\mathrm{CC}\left(\mathcal{P}_1^{\mathcal{B}}\right) \in o(\mathrm{CC}\left(\mathcal{P}_0^{\mathcal{B}}\right))$, in which case we say that $\mathcal{B}$ is "hard" for $\mathcal{P}_0$ and "easy" for $\mathcal{P}_1$.

This shows that our three necessary conditions for intrinsic universality are *really* necessary: no condition is stronger than any other.

4.4.1. A CA easy for PRED and hard for INV

The idea is to embed an equality test (more precisely, a *palindrom* test) launching signals invading the whole configuration, while keeping the prediction problem easy; see [8] or proposition 1 to see why this problem requires $\Omega(n)$ communicated bits. The idea is to use two components that both stay easy for PRED: one with tests that do not alter the component, and one with signals, moving quickly out of the way:

1. The first layer performs tests for equality, as described below, and initialy contains a word over the alphabet $\Gamma_1 = \{\overrightarrow{0}, \overrightarrow{1}, \overleftarrow{0}, \overleftarrow{1}, \top, \emptyset_1, K_1\}$. On figure 7, this layer is drawn with full lines.
   The dynamic of the first layer is simple : $\overrightarrow{a}$ states shift right, and $\overleftarrow{a}$ states shift left. $\top$ states do not move, and $\emptyset_1$ are spreading.

2. A layer with an automaton invading the configuration from a seed. We need five states on this layer: $\Gamma_2 = \{s, \emptyset_2, \rightarrow, \leftarrow, K_2\}$. We describe the rule below. On figure 7, this layer is drawn dashed.
   The rule here is even simpler: $\emptyset_2$ states do not move, $\rightarrow$ states shift right, $\leftarrow$ states shift left. State s represents a signal "seed", meaning that if it appears once, it disappears on the next step, and changes into a $\rightarrow$ signal on its right, and a $\leftarrow$ signal on its left.

We add a few rules that allow to verify the well-formedness of configurations. This allows us to ensure that there can be only one $\top$ state on the first layer, and that signals on the second layer never cross. States K_1 and K_2 are used for this purpose: if one of them appears somewhere, they *both* spread on both layers, thus erasing the whole configuration: the PRED problem becomes trivial.

- If a $\overleftarrow{a}$ state is found immediately next to an $\overrightarrow{a}$ state, then K_1 and K_2 are both raised.

- If a $\rightarrow$ signal is found in the same cell as an $\overrightarrow{a}$, or a $\leftarrow$ in the same cell as an $\overleftarrow{a}$, then K_1 and K_2 are raised. This ensures that signals on the second layer never cross.

Moreover, we introduce another rule to perform the equality test: when the test is negative (i.e. a $\top$ state has an $\overrightarrow{x}$ on its left, a $\overleftarrow{y}$ on its right, and $x \neq y$), then we place an s state on the second layer :
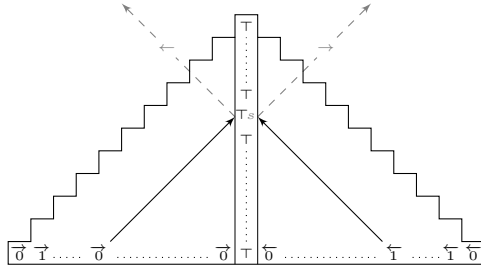
Figure 7: A CA easy for PRED and hard for INV

$$F\left(\begin{array}{c}\emptyset \\ \overrightarrow{a}\end{array}, \begin{array}{c}\emptyset \\ \top\end{array}, \begin{array}{c}\emptyset \\ \overleftarrow{a}\end{array}\right) = \begin{array}{c}\emptyset \\ \top\end{array}$$

$$F\left(\begin{array}{c}\emptyset \\ \overrightarrow{a}\end{array}, \begin{array}{c}\emptyset \\ \top\end{array}, \begin{array}{c}\emptyset \\ \overleftarrow{1-a}\end{array}\right) = \begin{array}{c}s \\ \top\end{array}$$

Proposition 10. *The CA F described above is such that:*

1. $\mathrm{CC}\,(\mathrm{PRED}_F) \in O(1)$,
2. *there is u such that $\mathrm{CC}\,(\mathrm{INV}_F^u) \in \Omega(n)$.*

Proof.   1. A protocol for PRED needs to predict the content of both layers: if the configuration is not well-formed, then a K_i state will appear somewhere and this is easy (and it can be checked locally by Alice and Bob). Else :

   - On the first layer, the result will always be the result of a shift if the initial configuration contains only $\overrightarrow{x}$ or $\overleftarrow{x}$ states, or if the $\top$ state is not the central cell of the configuration, and a $\top$ state else. This requires a constant number of communicated bits.

   - On the second layer, there are four – possibly overlapping – possibilities:

     – If the leftmost state of Alice's differs from the rightmost state of Bob's, and the central cell is a $\top$ state, the result is an s.
     – If the $\top$ state is not the central cell, but somewhere else in the left part, and the corresponding word is not a palindrom, then a $\rightarrow$ is launched (see figure 8).
     – If the initial configuration contained an s or a $\rightarrow$ in its leftmost cell, a $\rightarrow$ arrives to the top of the triangle.
     – Else, the result is a $\emptyset_2$.

   All of these can be checked locally and communicated between Alice and Bob within a constant number of bits.

2. Now we need to find a set of hard instances for the INV problem: with a background word u, with $\emptyset_j$ on both layers, and an initial configurations of the form $(\overrightarrow{0}, \overrightarrow{1})^n \top (\overleftarrow{0}, \overleftarrow{1})^n$ on the first layer, and $\emptyset^*$ on the second, we reduce the equality problem to INV.
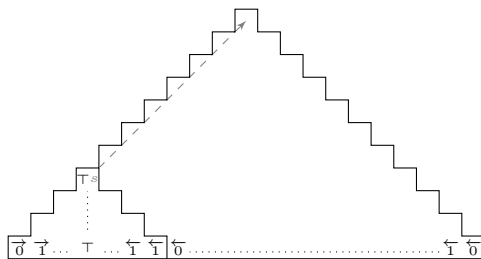
$\square$

18

Figure 8: A CA easy for Pred and hard for Inv

4.4.2. A CA easy for Cycle and hard for Inv

We can reuse the construction of paragraph 4.4.1: we already know that it is hard for Inv. What we need to do is to modify the rule so that on the second layer, when a $\rightarrow$ signal crosses a $\leftarrow$ signal, they both disappear and the resulting state is a $\emptyset_2$. This ensures that on cyclic configurations, even if signals are "raised" somewhere, they are "caught" by the cyclicity. The rest of the discussion is essentially the same as in paragraph 4.4.1, and we can conclude easily that the orbits of configurations containing at least one $\top$, or of ill-formed configurations, are always 1-periodic; the Cycle problem can be decided with no communication. In all other cases, the dynamic is nothing more than a shift: the protocol from 5 can be used.

4.4.3. A CA easy for Pred, and hard for Cycle

We can use once again (and for the last time) quite the same construction as in paragraph 4.4.1. We modify it to launch only one signal (in only one direction) when an error appears. Thus, as proven in section 4.4.1, the Pred problem remains easy. Now we need to prove that the Cycle problem is hard, but for this we can choose the instances on purpose.

If no test fails, the configuration will be 1-periodic: When all the tests have been done, the configuration is uniformly empty, except for the $\top$ states, and then nothing more happens. Otherwise, a signal will be launched. We need to show that the period of the configuration is then in $\Omega(n)$. But we can notice that a contiguous portion of $\Omega(n)$ cells can not have any signal (see Figure 9). Therefore, the period of the configuration is $\Omega(n)$ if and only if an error occurs.

4.4.4. A CA easy for Inv and hard for Cycle

As promised in remark 4.2, we now prove a protocol for the Inv problem of the rule described there:

Proposition 11. *The CA F described in the proof of proposition 9 is such that:*

$$\forall u, \mathrm{CC}\left(\mathrm{Inv}_F^u\right) \in O(1)$$

Proof. Let u be any word over the alphabet for F. First, if the orbit of p_u contains a spreading state, then $p_u(w)$ quickly becomes uniform with the spreading state everywhere, independently from w. Else, the discussion is a little more subtle. Let us note the periodic background $p_u = (p_{u_1}, p_{u_2}, p_{u_3})$, and let $w = (w_1, w_2, w_3)$ the input, split between Alice and Bob.
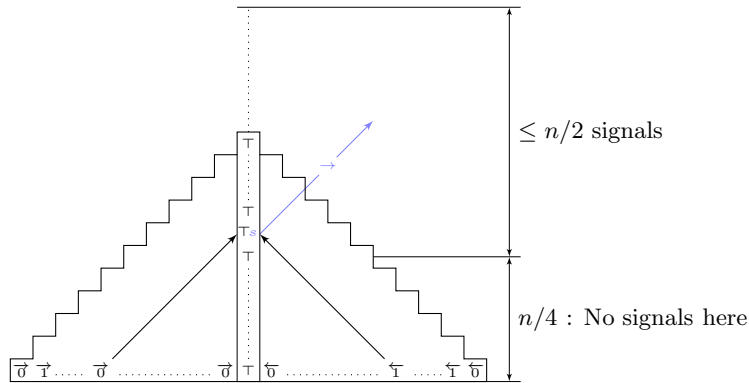
Figure 9: A CAeasy for PRED and hard for CYCLE.

1. If $p_{u_1}(w_1) \neq p_{u_1}$, and $p_{u_2}(w_2) \neq p_{u_2}$, and then either a spreading state is generated, or the differences on components one and two are shifted in opposite directions, thus also invading p_u.

2. If $p_{u_1}(w_1) = p_{u_1}$ and $p_{u_2}(w_2) = p_{u_2}$, maybe the third component (the actual "tests") changes between p_u and $p_u(w)$, but then there is an easy way to transmit whole configurations : Alice can simply tell Bob that her part is the same as in p_u, on the first two components. If Bob does the same, then both know both "sets", and they can check without more communication if their respective portions of $p_{u_3}(w_3)$ ever generates a spreading state : if so, $p_u(w)$ is invaded, else it is not.

3. Else, without loss of generality, we can assume that $p_{u_1}(w_1) = p_{u_1}$ and $p_{u_2}(w_2) \neq p_{u_2}$. There are two cases :

   • Either $p_{u_3}(w_3) = p_{u_3}$ (the "tests" are the same in p_u and $p_u(w)$), and then using the trick from (2), Alice and Bob can know $p_{u_1}(w_1)$ and $p_{u_3}(w_3)$ completely, within constant communication.
   Then, since they each know a part of set $p_{u_2}(w_2)$, and they both know $p_{u_3}(w_3)$, they can check disjointness with $p_{u_1}(w_1)$ separately and tell if a spreading state ever appears, which is the only way $p_u(w)$ can be invaded in this case.

   • If $p_{u_3}(w_3) \neq p_{u_3}$, then either a spreading state is generated, or $p_{u_3}(w_3)$ stays fixed, and $p_{u_2}(w_2)$ shifts to infinity: in both cases, $p_u(w)$ is invaded.

$\square$

4.4.5. A CA easy for INV *and hard for* PRED

Elementary rule 218 is a natural example exhibiting this property. Unfortunately, the proof is quite technical and requires an in-depth study of rule 218, which we chose to delay until section 6.1, for conciseness of this –already long– section, and consistency of section 6.

20

4.4.6. An CA easy for CYCLE *and hard for* PRED

We describe the natural example of Rule 33 in Section 6.3, which has a protocol in constant time for CYCLE, and for which any deterministic protocol for PRED is in $\Omega(\log n)$.

5. Intrinsic universality: Ruling out complex CA

Here we show that for two of our canonical problems – namely, PRED and INV – we were able to find a CA of maximal algorithmic complexity (*complete*), and yet very simple with respect to our framework.

More precisely, we are going to show that, for problems PRED and INV, there exists a CA F for which the communication complexity of the problem is low while its classical computational complexity is the highest one can expect.

Therefore, we are ruling out such non-trivial CA from being intrinsically universal.

5.1. Prediction

T. Neary and D. Woods proved "the P-completeness of Rule 110" [11]. In our language, they proved that the problem $\text{PRED}_{F_{110}}$ is P-complete. A very natural question arises: What do classical algorithmic properties of CA, such as P-completeness, imply on their communication complexity counterpart?

As we show in this section, such a strong computational property is not enough to guarantee maximal communication complexity. However, we do not know of an automaton that would have, for instance, polylogarithmic communication complexity, and still a P-complete prediction problem, nor do we have a nonexistence proof. We leave this as an open problem.

Proposition 12. *For any $k \geq 1$, there exists a CA F such that*

$$\text{CC}\,(\text{PRED}_F) \in O(n^{1/k})$$

and PRED_F is P-complete.

Proof. Let $\mathcal{M}$ a Turing machine. We construct a CA F simulating $\mathcal{M}$ slowly but still in polynomial time: it takes n^k steps of F to simulates n steps of $\mathcal{M}$. Hence, by a suitable choice of $\mathcal{M}$, the problem of predicting F is P-complete.

First it is easy to construct a CA simulating $\mathcal{M}$ in real time. We encode each symbol of the tape alphabet of the Turing machine by a CA state, and add a "layer" for the head, with '→' symbols on its left and '←' symbols on its right. We guarantee this way that there can be only one head: if a '→' state is adjacent to a '←' state without a head between them, we propagate a spreading "error" state destroying everything.

We then add a new layer to slow down the simulation: it consists in a single particle (we use the same trick to ensure that there is only one particle) moving left and right inside a marked region of the configuration. More precisely, it goes right until it reaches the end of the marked region, then it adds a marked cell at the end and starts to move left to reach the other end, doing the same thing forever. Clearly, for any cell in a finite marked region, seeing n traversals of the particle takes $\Omega(n^2)$ steps. Then, the idea is to authorize head moves, in the previous construction, only at particle traversals. This way, n steps of

$\mathcal{M}$ require n^2 time steps of the automaton. By adding another particle layer, one can also slow down the above particle with the same principle and it is not difficult to finally construct a CA F such that n steps of $\mathcal{M}$ require n^k time steps of F. We have represented in Figure 10 the behavior of the particle, with the dashed arrow representing a Turing transition.

Now if the initial configuration does not respect the rules described above, then a spreading error state is generated and Alice and Bob can notice it within constant communication. In all other cases, it is enough for Alice or Bob to know the value of all the $2 \cdot n^{1/k}$ states around the initial position of the head, because the computation of the Turing machine simply does not depend on the rest of the initial configuration. So for these cases, at most $n^{1/k}$ bits need to be communicated for Alice or Bob to compute the answer. Note that if the bounds for the particle are absent from the initial configuration, then no transition can happen, thus Alice and Bob know the result in constant time.
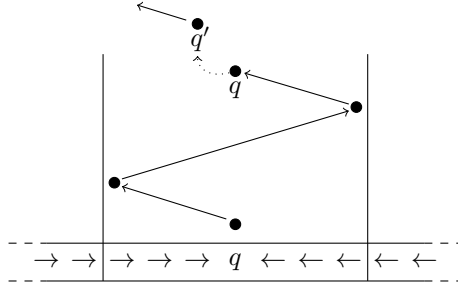

Figure 10: A CA for which PRED is P-complete.

$\square$

Remark. *A result by Hromkovic (see [2]) states that a Turing machine with a single head working in time $t(n)$ can only recognize a language of communication complexity less than $O(\sqrt{t(n)})$. Said differently, a CA simulating a Turing machine cannot produce instances of communication complexity more than $O(\sqrt{n})$ for the prediction problem on configurations with a single head (whatever the machine does).*

5.2. Invasion

This problem is even more complex than PRED: It is in fact undecidable. However, since there is no limitation on the "classical" computational power of Alice and Bob, it can still be decided within very little communication.

Proposition 13.

1. *For any CA F and any word u, we have: $\mathrm{INV}_F^u \in \Pi_1^0$.*
2. *Their exist F and u such that INV_F^u is Π_1^0-complete, and yet $\mathrm{CC}\left(\mathrm{INV}_F^u\right) \in O(\log n)$*

Proof.

1. Let F and u be fixed and consider the problem INV_F^u. Given an input $x_1, \dots, x_n$, we use the notations $\delta_l(t)$ and $\delta_r(t)$ for the leftmost and righmost differences at time t between the orbit of p_u and the orbit of $p_u(x_1 \cdots x_n)$ as in Definition 4.

Claim. *There exists a recursive function β such that for any n, any input $x_1, \ldots, x_n$ and any $\Delta \geq 0$ we have:*

$$\exists t,\ \delta_r(t) - \delta_l(t) \geq \Delta \iff \exists t \leq \beta(\Delta),\ \delta_r(t) - \delta_l(t) \geq \Delta.$$

The proof follows from the above claim because the invasion problem can be expressed as the following Π_1^0 predicate:

$$\forall \Delta \geq 0,\ \underbrace{\exists t \leq \beta(\Delta),\ \delta_r(t) - \delta_l(t) \geq \Delta}_{\text{recursive predicate}}$$

Proof of the claim. First, the orbit of p_u is ultimately periodic: There are t_0 and p such that for any $t \geq t_0$ we have $F^t(p_u) = F^{t+p}(p_u)$. Given an input $x_1, \ldots, x_n$ of the problem, denote by $w(t)$ the word of length $\delta_r(t) - \delta_l(t)$ starting at position $\delta_l(t)$ in configuration $F^t\big(p_u(x_1, \ldots, x_n)\big)$. The key point is that for any $t \geq t_0$, the triple

$$\chi(t+1) = \big(w(t+1), \delta_l(t+1) \bmod |u|, t+1 \bmod p\big)$$

is uniquely determined by the triple

$$\chi(t) = \big(w(t), \delta_l(t) \bmod |u|, t \bmod p\big)$$

(because the word $w(t)$ "evolves" in a periodic context and knowing the offset of the position of $w(t)$ in that context is enough to know $w(t+1)$). Therefore, if the words $w(t)$ are bounded by Δ for a sufficiently long time (exponential in Δ), then the triple $\chi(t)$ will take a value already taken before and the sequence $\big(\chi(t)\big)_t$ will be ultimately periodic, showing that $|w(t)|$ is bounded and that there is no invasion. Adding t_0 to this exponential function is a convenient choice for β. $\qquad\square$

2. We build a CA F that simulates a 2-counter machine [10]. More precisely, standard states have two layers: a data layer over states $A, M, B, 0$, used to store the value of the 2 unary counters, and a control layer made of a Turing head storing a state from Q, with the extra $\rightarrow$ and $\leftarrow$ symbols ensuring the uniqueness of the head. Finally, F possesses a blank state $\emptyset$ and a spreading state K to deal with encoding problems. The state set is therefore

$$\big(Q \cup \{K, \emptyset, \rightarrow, \leftarrow\}\big) \times \{A, B, 0, M\}.$$

A valid configuration is a configuration everywhere equal to $\emptyset$ except on finite coding segments which have the folloing form (see figure 11):

- the data layer must be of the form: $0^* A^+ M B^+ 0^*$;
- the control layer must be of the form: $\rightarrow^+ q \leftarrow^+$ with $q \in Q$.

Control layer	$\emptyset$	$\emptyset$	$\rightarrow$	$\rightarrow$	$\rightarrow$	q	$\leftarrow$	$\leftarrow$	$\emptyset$	$\emptyset$	$\emptyset$
Data layer	$\emptyset$	$\emptyset$	A	A	A	M	B	0	$\emptyset$	$\emptyset$	$\emptyset$

Figure 11: A well-formed piece of configuration. The counter A contains value 3 and the counter B contains value 1 in this example.

The number of As and Bs represent the current value of the 2 counters. The behaviour of F is the following:

23

- If the configuration is not valid (which can be detected locally), then the state K is generated and spreads;

- If the configuration is valid, then on each coding segment, the (necessarily unique) head goes repeatedly from one end of the segment to the other end, and extends the segment at each pass by adding a $\rightarrow$ on the left (resp. $\leftarrow$ on the right) and a 0 on the data layer. If the extension step is blocked by another segment, then the state K is generated and spreads;

- Moreover, at each pass on the segment, the head executes one of the basic 2-counter machine's instructions:
  - testing if a counter is empty can be done by checking if there is a 0 on the right (resp. the left) of the unique M;
  - decrementing can be done be replacing the leftmost A (resp. rightmost B) by a 0;
  - incrementing can be done by replacing a 0 by A on the left of the leftmost A (resp. by B on the right of the rightmost B); there must be a 0, because the segment is extended at each passage by both sides;
  - finally, the head can simply stop.

If any order given to the head leads to an incoherence (decrement an empty counter, write a B when on the 'A' part of the segment, etc), the state K is genereated and spreads.

With this definition, and if $u = \emptyset$, the halting problem for the 2-counter machine encoded in F (input: value of counters; output: does it halt started from these values ?) clearly reduces to INV_F^u (halt $\Longleftrightarrow$ no invasion). Therefore, by a suitable choice of the 2-counter machine used to construct F, we have that INV_F^u is Π_1^0-complete.

To conclude the proof, we show that $\text{CC}(\text{INV}_F^u) \in O(\log(n))$. Given an input w split between Alice and Bob, the following protocol determines whether $\text{INV}_F^u(w) = 1$:

- first Alice and Bob check whether the input configuration is valid; if not, the answer is 'invasion'; this can be done with $O(1)$ bits of communication since validity is a local property;

- the configuration being valid, Alice and Bob communicate so that for any pair of consecutive valid segments s_1 and s_2, either Alice or Bob knows the state of both s_1 and s_2 and the distance between them; to achieve this, even if a segment is split between Alice's part and Bob's part, it is sufficient that they communicate $O(\log(n))$ bits; indeed, a segment is completely defined by:
  - the value and position of the head,
  - number of 0 states on the right and the same on the left,
  - number of As and number of Bs.

- since for each pair of valid segment, Alice or Bob as enough information to detect a possible future collision, they can determine together with $O(1)$ bits of communication whether there is invasion or not; indeed, invasion is equivalent to: either their is a collision somewhere, or their is a single segment holding a non-halting computation.

$\square$

5.3. Cycle-length

For this problem, we could find a CA of maximal algorithmic complexity, as shown by the following proposition. However, we have to leave as an open problem the existence of a CA F for which both CYCLE_F^k is PSPACE-complete for some $k \in \mathbb{N}$, and $\text{CC}\left(\text{CYCLE}_F^k\right) \in o(n)$.

Proposition 14.   1. *For any CA F and any $k \geq 1$, $\text{CYCLE}_F^k \in$ PSPACE.*
     2. *Their exist F and k such that CYCLE_F^k is PSPACE-complete.*

Proof.

1. Let F and $k \geq 1$ be fixed. The length of the cycle reached by iterating F on a periodic initial configuration c can be determined in polynomial space with the algorithm described below. Let n be the period of c. Starting from c, the cycle is reached in less than α^n steps where α is the cardinal of the state set.
   (a) compute $c_0 = F^{\alpha^n}(c)$ (memory usage: $O(n)$);
   (b) memorize c_0 and compute the first t such that $F^t(c_0) = c_0$ (memory usage: $O(n)$ because such a t is less than α^n).
2. To show this, we embed a Turing machine $\mathcal{M}$, deciding a PSPACE-complete language, in a cyclic configuration for a cellular automaton. $\mathcal{M}$ works in polynomial space, meaning that there is a polynomial $P \in \mathbb{N}[X]$ such that for any $x \in \Gamma^*$, it will never use more than $P(|x|)$ tape cells.
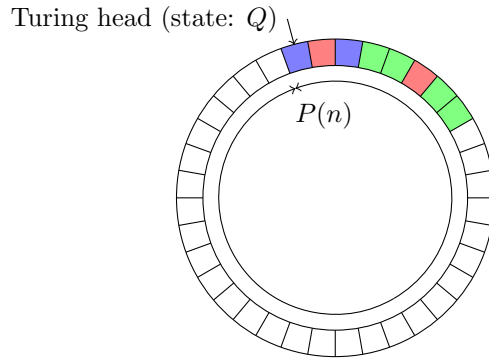

Turing head (state: Q)

$P(n)$

Figure 12: The output of the transducer used in Proposition 14.

We can encode a Turing machine easily into a simple cellular automaton F: the states code for the Turing tape cells, and there is a special "head" state carrying the state of the machine. It can be easily shown that we can encode the transitions of a Turing machine into a local cellular automaton rule, ensuring that if there is only one head at the beginning, then it will be so during all the computation.

Moreover, the accepting state is *spreading*, meaning that if it appears somewhere, it spreads over all the configuration in both directions. The rejecting state launches a particle erasing the configuration (i.e., writing blank states everywhere), but shifting clockwise. In this way, an accepting computation will result in period 1, whereas rejecting computations will yield periods of the size of the configuration.

A polynomial-time transducer can easily encode an input x for $\mathcal{M}$ into a (cyclic) configuration of F, like shown in figure 12. It first directly translates x into states of F, then computes $P(|x|)$ and outputs $P(x)$ blank states.
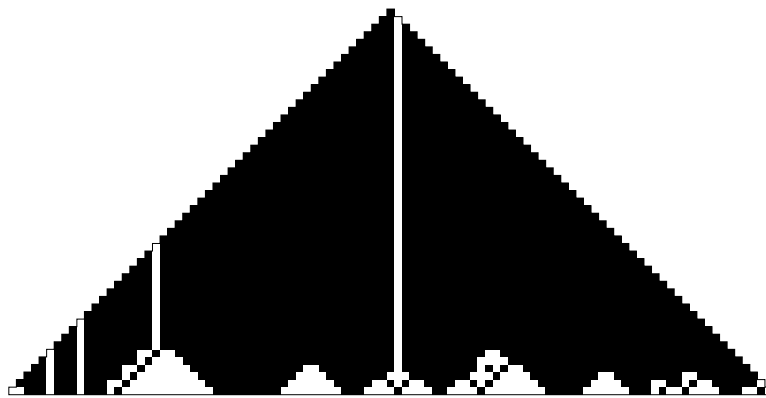
$\square$

6. Intrinsic universality: Ruling out concrete elementary CA

6.1. CA Rule 218

The local function $f_{218} : \{0,1\}^3 \rightarrow \{0,1\}$ of CA rule 218 is defined in Figure 13(a).

0	1	0	1	1	0	1	1
0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1

(a) F_{218}.

(b) Example of a space-time diagram for CA Rule 218.

Figure 13: CA rule 218.

From the result of [5] we already knew that $\mathrm{CC}\left(\mathrm{PRED}_{F_{218}}\right) \in O(\log(n))$. It follows from Corollary 2 that Rule 218 is not intrinsically universal. Nevertheless, the proof of [5] was very long and complicated. As we are going to see now, the invasion approach gives a short and elegant proof of the same result.

Definition 7. *A word is* additive *if 1s are isolated and separated by an odd number of 0s. By extension, an infinite configuration is* additive *if it contains only additive words.*

Lemma 2. *Additivity is preserved by iterations. Moreover, if abc is additive then:*

$$f_{218}(a,b,c) \neq f_{218}(1-a,b,c) \text{ and } f_{218}(a,b,c) \neq f_{218}(a,b,1-c).$$

Proof. First additivity is preserved by iterations because 010^n10 becomes $010^{n-2}10$ for $n \geq 3$ and 01010 becomes 000.

To conclude the lemma, it is sufficient to check that, for any a,b,c such that 11 is not a factor of abc then:

$$f_{218}(a,b,c) \neq f_{218}(1-a,b,c) \text{ and } f_{218}(a,b,c) \neq f_{218}(a,b,1-c).$$

$\square$

Lemma 3. *Let c be any non-additive configuration. Then, after a finite time, the word 11 appears in the evolution and this word is a wall.*

Proof. First 11 is a wall because:

$$f_{218}(*,1,1) = f_{218}(1,1,*) = 1.$$

To conclude it is sufficient to check that the image of $10^n 1$ with $n \geq 2$ is $10^{n-2}1$.

$\square$

Proposition 15. *For all u, we have $\mathrm{CC}\left(\mathrm{INV}_{F_{218}}^u\right) \leq 1$.*

Proof. First, if the configuration p_u is non-additive then, by Lemma 3, at some time t a wall appears periodically in $F_{218}^t(p_u)$. Hence, for any $x_1, \ldots, x_n$, the differences between $p_u(x_1, \ldots, x_n)$ and p_u are bounded to a fixed finite region. Said differently, there is never propagation for such an u.

Now consider the case where p_u is additive. By Lemma 2, we have for any $x_1, \ldots, x_n$:

- either $p_u = p_u(x_1, \ldots, x_n)$,
- or for any $t \geq 0$:

$$\delta_l(t) = \delta_l(0) - t$$
$$\delta_r(t) = \delta_r(0) + t$$

Therefore, the problem consists in deciding whether p_u and $p_u(x_1, \ldots, x_n)$ are equal, which can be done with 1 bit of communication. $\square$

Corollary 4. *CA Rule 218 is not intrinsically universal.*

As promised in section 4.4.5, it remains to show that the deterministic (possibly with several rounds) communication complexity of the PRED problem for rule 218 is "hard", according to our conventions:

Proposition 16.
$$\mathrm{PRED}_{F_{218}} \in \Omega(\log n)$$

Proof. To show this, we construct a fooling set S_n (see Definition 1 or [8]):

$$S_n = \{(1^{n-k}0^k, 0^{k+1}1^{n-k}, 0 \leq k \leq n\}$$

We show that S_n is a fooling set for Rule 218: In fact, on all configurations of the form $1^{n-k}0^{2k+1}1^{n-k}$, the result of $\mathrm{PRED}_{F_{218}}$ is always 0. On configurations of the form $1^{n-i}0^{i+j+1}1^{n-j}$ where $i \neq j$, it is always 1. This can be easily shown from the collection of lemmas of [5], and we illustrate it on Figure 14. Thus, since $|S_n| = n+1$, we deduce that a deterministic protocol solving $\mathrm{PRED}_{F_{218}^n}$ can not take less than $\log(n+1)$ steps:

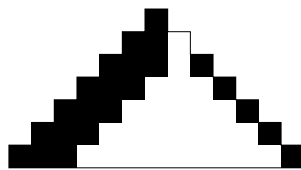$$\mathrm{CC}\left(\mathrm{PRED}_{F_{218}^n}\right) \in \Omega(\log(n))$$

$\square$

Figure 14: A configuration of the fooling set S_n for rule 218

6.2. CA Rule 94

The local function $f_{94} : \{0,1\}^3 \to \{0,1\}$ of CA Rule 94 is defined in Figure 15(a).

0	1	1	1	1	0	1	0
0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1

(a) f_{94}.

(b) Example of a space-time diagram for CA Rule 94.

Figure 15: CA Rule 94.

Here appears clearly how powerful the invasion approach is (as a tool for proving non-universality). Finding an upper bound (a protocol) for $\mathrm{CC}\left(\mathrm{PRED}_{F_{94}}\right)$ seems to be hard. Nevertheless, here we prove in a rather simple way that its invasion complexity is logarithmic.

Definition 8. *A configuration is* additive *if its language is included in* $\left((00)^+(11)^+\right)^*$ *(blocks of 0s or 1s are always of even length).*

Lemma 4. *f_{94} is bi-permutative when restricted to additive configurations (it behaves like f_{90}) and additive configurations are stable under iterations.*

Proof. For stability of additive configurations, it is sufficient to check that $00(11)^n00$ becomes $11(00)^{n-1}11$ and $11(00)^n11$ becomes $11(00)^{n-1}11$ for $n \geq 1$.
    f_{94} differs from f_{90} only for transition 010, hence bi-permutativity. $\square$

Lemma 5. *If c is a non-additive configuration which does not contain* 010*, then* 101 *appears after a finite time and it is a wall. More precisely, a wall appears after $t + 1$ steps of CA Rule 94 at the middle of any occurrence of $10^{2t+1}1$ or $01^{2t+3}0$ (with $t \geq 0$).*

28

Proof. First 101 is stable under iterations of f_{94}. Second, 10^n1 with $n \geq 2$ is sent to $10^{n-2}1$ and 01^n0 is sent to $10^{n-2}1$ for $n \geq 2$. $\square$

Lemma 6. *The orbit of a configuration c contains a wall if and only if $F_{94}(c)$ is not additive.*

Proof. From Lemma 5, it is enough to show that if c is a configuration not containing 101 then $F_{94}(c)$ does not contain 010. For that, it is sufficient to check that any word u such that $f_{94}(u) = 010$ must contain 101. $\square$

From the 2 lemmas above, we get the following proposition.

Proposition 17. *For any u we have* $\mathrm{CC}\left(\mathrm{INV}_{F_{94}}^u\right) \in O(\log(n))$.

Proof. If u is such that the orbit of p_u contains a wall, then invasion never occurs.

If u is such that the orbit of p_u does not contain any wall, then it means that $F_{94}(p_u)$ is additive (by Lemma 6). In this situation, two cases are to be considered depending on the input $x_1, \ldots, x_n$. Knowing in which case we are can be done within constant number of bits:

- either $F_{94}(p_u(x_1, \ldots, x_n))$ is also additive and then, by Lemma 4, there is invasion if and only if $F_{94}(p_u) = F_{94}(p_u(x_1, \ldots, x_n))$. This can be decided with a finite number of bits of communication.

- or $F_{94}(p_u(x_1, \ldots, x_n))$ is not additive. Then it contains some $10^{2t+1}1$ or some $01^{2t+3}0$ (with $t \geq 0$) because, as shown in the proof of lemma, if the image of a configuration contains 010, then it must also contain 101. Consider the leftmost and the rightmost occurrences of this kind of words. Since walls appear above the middle of these two occurrences after a time equal to half their lengths (Lemma 5), the fact there is invasion or not does not depend on what is beetween that two occurrences. It takes $O(\log(n))$ bits of communication for Alice to know the positions of these two occurrences and the exact words present at their positions (of type $10^{2t+1}1$ or $01^{2t+3}0$). Moreover, as soon as Alice knows this she also knows that on the left of the leftmost occurrence and on the right of the rightmost occurrence, the configuration is additive. If there is one difference with p_u in those additive parts, then there is invasion. If not, then Alice has got enough information to decide invasion. Deciding in which of the two cases we are can be done within constant communication.

$\square$

Corollary 5. *CA Rule 94 is not intrinsically universal.*
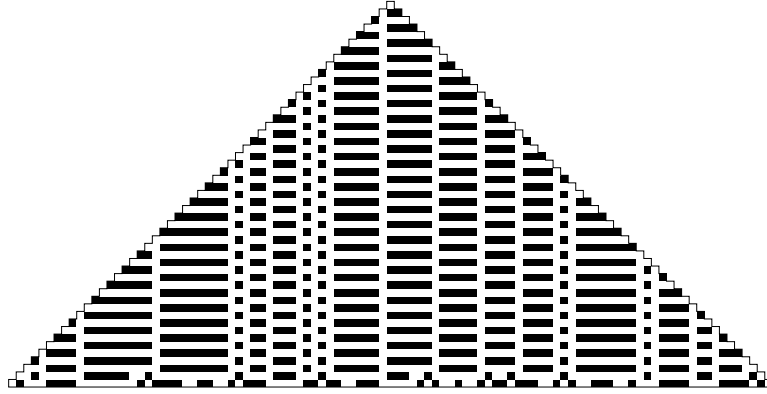
6.3. CA Rule 33

We are going to show that this rule, although non-trivial for the PRED problem, needs zero communication for the CYCLE problem. To show this, we prove that the cycle length of Rule 33 is always 2. The local function $f_{33} : \{0,1\}^3 \rightarrow \{0,1\}$ of CA Rule 33 is defined in Figure 16(a).

Lemma 7. *All configurations that do not contain neither 101 (isolated 0s) nor 1001 (isolated 00s) are stable under $(F_{33})^2$.*

1	0	0	0	0	1	0	0
0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1

(a) f_{33}.

(b) Example of a space-time diagram for CA Rule 33.

Figure 16: CA Rule 33.

Proof. We call A_0 the set of configurations without isolated 0s, and A_{00} the set configuration without isolated 00s. First notice that the only antecedent of 101 is 10101, which contains an isolated 0, thus A_0 is stable under F_{33}. With an exhaustive exploration of all configurations of the form $u = abxyzcd$ where $xyz \in \{000\ldots111\}$, and $u \in A_0 \cap A_{00}$, we observe that:

$$\forall u \in A_0 \cap A_{00}, |u| = 7, (F_{33})^2(u_1 \ldots u_7) = u_3 u_4 u_5$$

$\square$

Lemma 8. *All (cyclic) configurations of length n, different from $(01)^{\lfloor n/2 \rfloor}$, do not contain isolated 0s after $\left\lfloor \frac{n}{2} \right\rfloor$ steps of CA Rule 33.*

Proof. We already noticed in Lemma 7 that the only possible antecedent of 101 is 10101. Thus, there can be an isolated 0 after $\left\lfloor \frac{n}{2} \right\rfloor$ steps only if there are at least $\left\lfloor \frac{n}{2} \right\rfloor$ isolated 0s in the initial configuration, i.e. if the initial configuration is $(01)^{\lfloor n/2 \rfloor}$.

$\square$

Corollary 6. *After $\left\lfloor \frac{n}{2} \right\rfloor + 1$ steps of CA Rule 33, there are no isolated couples of 0s.*

Proof. The only antecedents of 1001 contain an isolated 0.

$\square$

Corollary 7. *After $\left\lfloor \frac{n}{2} \right\rfloor + 1$ steps, CA Rule 33 becomes periodic, with period 2.*

Proposition 18.
$$\mathrm{CC}\left(\mathrm{PRED}_{F_{33}}\right) \in \Omega(\log n)$$

Proof. As usual, we just find a fooling set (see Definition 1). Consider the following set S_n:

$$S_n = \{(1^{n-2k}(01)^k 0, (10)^k 1^{n-2k}), 0 \le k \le \lfloor n/2 \rfloor\}$$

30

It can be easily verified that:

$$\begin{cases} F_{33}^n(1^{2n-k}(01)^k 0(10)^k 1^{2n-k}) & = & n \mod 2 \\ F_{33}^n(1^{2n-i}(01)^i 0(10)^j 1^{2n-j}) & = & 1 + (n \mod 2) & \text{whenever } i \neq j \end{cases}$$

Since $|S_n| = \lfloor \frac{n}{2} \rfloor$, we conclude that a deterministic protocol for predicting rule 33 needs $\Omega(\log n)$ bits of communication.

□

7. Conclusion

We have suggested a method to prove negative results concerning intrinsic universality in CA. We have shown that this approach can be used both to show that global dynamical properties can imply non-universality, and to rule out some concrete cellular automata from being universal. We believe that this work should go on in the following directions:

- It seems that more can be said about the communication complexity problems for the class of surjective CA and some of its sub-classes (k-to-1, d-separated, left/right-closing, etc. [6]);

- The case of elementary rules 218 and 94 shows that low-cost communication protocols can be found in CA that are not linear, but containing a linear component 'in competition' with another component. Finding a general formalisation for such kind of behaviours could be useful to treat many other concrete examples.

- Concerning concrete CA, ruling out as many elementary rules as possible from being intrinsically universal seems to be an interesting (but ambitious) goal. We could also consider other natural classes of small CA (one-way automata, totalistic rules, etc.).

- The splitting of inputs that induce maximal communication complexity is a key parameter, especially for the prediction problem. There is no reason for such maximal splittings to be unique, and if it is unique, there is no reason to be located in the middle of the input. We suspect that there are some links between directional entropy and the evolution of such maximal splitting (when increasing the input size).

- Although completely formalized in dimension 1, there is no doubt that this approach can be adapted to higher dimensions; it could be the occasion to adopt other communication complexity models (like the multiparty model) and discuss other ways of splitting the input.

[1] Laurent Boyer and Guillaume Theyssier. On local symmetries and universality in cellular automata. In *STACS 2009*, pages 195–206, 2009.

[2] Juraj Hromkovič and Georg Schnitger. Communication complexity and sequential compuation. In *MFCS '97*, pages 71–84, London, UK, 1997. Springer-Verlag.

[3] B. Durand and Z. Róka. *Cellular Automata: a Parallel Model*, volume 460 of *Mathematics and its Applications*, chapter The game of life: universality revisited, pages 51–74. Kluwer Academic Publishers, 1999.

[4] Christoph Dürr, Ivan Rapaport, and Guillaume Theyssier. Cellular automata and communication complexity. *TCS*, 322(2):355–368, 2004.

[5] Eric Goles, Cedric Little, and Ivan Rapaport. Understanding a non-trivial cellular automaton by finding its simplest underlying communication protocol. In *ISAAC 2008*, pages 71–94, 2008.

[6] G. A. Hedlund. Endomorphisms and automorphisms of the shift dynamical systems. *Mathematical Systems Theory*, 3(4):320–375, 1969.

[7] P. Kurka. Languages, equicontinuity and attractors in cellular automata. *Ergodic theory and dynamical systems*, 17:417–433, 1997.

[8] Eyal Kushilevitz and Noam Nisan. *Communication complexity*. Cambridge university press, 1997.

[9] Jacques Mazoyer and Ivan Rapaport. Inducing an order on cellular automata by a grouping operation. *Discrete Applied Mathematics*, 91:177–196, 1999.

[10] Marvin L. Minsky. *Computation: Finite and Infinite Machines*. Prentice-Hall, Englewood Cliffs, New Jersey, 1967.

[11] Turlough Neary and Damien Woods. P-completeness of cellular automaton Rule 110. In *In ICALP 2006, volume 4051 of LNCS*, pages 132–143. Springer, 2006.

[12] Nicolas Ollinger. Universalities in cellular automata: a (short) survey. In B. Durand, editor, *JAC'08*, pages 102–118. MCCME Publishing House, Moscow, 2008.

[13] Mathieu Sablik. Directional dynamics for cellular automata: A sensitivity to initial condition approach. *TCS*, 400(1-3):1–18, 2008.

[14] Guillaume Theyssier. *Cellular automata : a model of complexities*. PhD thesis, ENS Lyon, 2005.

[15] John von Neumann. *The theory of self-reproducing cellular automata*. University of Illinois Press, Urbana, Illinois, 1967.

[16] Andrew Chi-Chih Yao. Some complexity questions related to distributive computing (preliminary report). In *STOC*, pages 209–213. ACM, 1979.