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

Let F be a cellular automaton (CA). This paper establishes necessary conditions for F in order to be intrinsically universal. The central idea is to consider the communication complexity of various "canonical problems" related to the dynamics of F. We show that the intrinsic universality of F implies high communication complexity for each of the canonical problems. This result allows us to rule out many CAs from being intrinsically universal: The linear CAs, the expansive CAs, the reversible CAs and the elementary CAs 218, 33 and 94. The notion of intrinsic universality is related to a process by which we change the scale of space-time diagrams. Therefore, in this work we are answering pure dynamical question by using a computational theory. This communication complexity theory, on the other hand, provides a finer tool than the one given by classical computational complexity analysis. In fact, we prove that for two of the canonical problems there exists a CA for which the computational complexity is maximal (P-complete, or Π_1^0-complete) while the corresponding communication complexity is rather low. We also show the orthogonality of the problems. More precisely, for any pair of problems there exists a CA with low communication complexity for one but high communication complexity for the other.

Key words: cellular automata, communication complexity, intrinsic universality.

1. Introduction

Since the pioneering work of J. von Neumman [16], universality in cellular automata (CAs) has received a lot of attention (see [13] for a survey). Historically, the notion of universality used for CAs 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, 10, 13].

[*] Corresponding author (guillaume.theyssier@univ-savoie.fr)

[*] Partially supported by programs Fondap and Basal-CMM, Fondecyt 1070022 (E.G) and Fondecyt 1090156 (I.R.).

This definition of intrinsic universality –which relies on a notion of simulation formalized later in this paper– may seem very restrictive. Nevertheless, the intrinsic universality property can be very common in some natural families of CAs [1]. But, most importantly, by completely formalizing[2] the notion of universality, we facilitate the proof of negative results.

In fact, in this paper we will explain how to rule out particular elementary CAs from being intrinsically universal (notice that CA Rule 110 is Turing-universal [2] and no elementary CA in known to be intrinsically universal). More precisely, we are going to show that the tool of communication complexity seems to be a particularly good candidate in order to obtain such negative results.

The notion of intrinsic universality is related to a process by which we change the scale of space-time diagrams. Therefore, in this work we are answering pure dynamical question by using computational tools.

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 $\mathcal{P}_F$. We study the computational complexity of those problems showing that they are complete with respect to the class they belong (choosing the right F in every case).

In Section 4 we explain how to infer deep properties of F from the study of the communication complexity of $\mathcal{P}_F$. More precisely, we prove that if the communication complexity of any 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 CAs from being intrinsically universal. We conclude that linear, expansive and reversible CAs are not intrinsically universal. We also show the orthogonality of the canonical problems: For any pair of these problems there exists a CA with low communication complexity for one but high communication complexity for the other.

In Section 5 we explain clearly why the communication complexity approach appears to be a promising tool for ruling out CAs from being intrinsically universal. More precisely, we prove that for two of the canonical problems there exists a CA for which the decison version is hard (P-complete, Π_1^0-complete) while the communication complexity is rather low.

Finally, in Section 6 we use our results for proving that some concrete elementary CAs are not intrinsically universal. Although looking at several space-time diagrams of these CAs 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, to our knowledge, has been never been done before.

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

2. Basic definitions

2.1. Communication complexity

Communication complexity is a measure introduced by A. C.-C. Yao in [17], 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 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".

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 CAs.

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 depth of a protocol tree computing ϕ.

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 [9]).

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 [9]).

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 [9]).

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:*

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

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

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

The following lower bounds hold:

- $\mathbf{cc}(\phi_{\text{EQ}}) \geq n$.
- $\mathbf{cc}(\phi_{\text{IP}}) \geq n$.
- $\mathbf{cc}(\phi_{\text{DISJ}}) \geq n$.

2.2. Splitting the input of computational problems

Let us consider now classical computational input-output problems. In this work we only take into account problems $\mathcal{P} : Q^* \to Z$ 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 $\mathbf{cc}\big(\mathcal{P}|_n^i\big)$ 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. Putting all together, we associate to any problem a function as explained in the following definition.

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

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

2.3. Cellular automata

In this paper we are always going to consider one-dimensional CAs. 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 problems 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 CAs: 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.

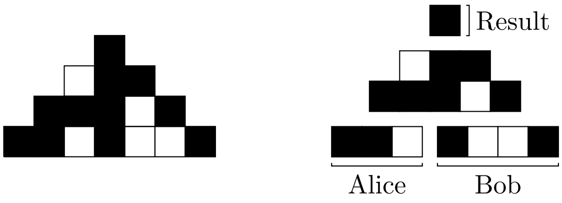
Definition 3. *Let F be a CA. The problem* $\textsc{Pred}_F : Q^* \to Q$ *is defined as follows:*

$$\textsc{Pred}_F(u) = \big(f^*(u)\big)_1,$$

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 $2r$.

We could ask the classical algorithmic complexity questions such as the decidability of $\textsc{Pred}_F$, $\textsc{time}(\textsc{Pred}_F)$ or $\textsc{space}(\textsc{Pred}_F)$. In this particular case it is clear that $\textsc{time}(\textsc{Pred}_F) \in O(n^2)$.

As we have already said before, we can also view $\textsc{Pred}_F$ 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).



(a) A space-time diagram of Rule 110.

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

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

More precisely, for every $1 \leq i \leq (n-1)$, $\textsc{Pred}_F|^i_n : Q^i \times Q^{n-i} \to Q$ is such that $\textsc{Pred}_F|^i_n(x, y) = (f^*(xy))_1$. This function $\textsc{Pred}_F|^i_n$ 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).

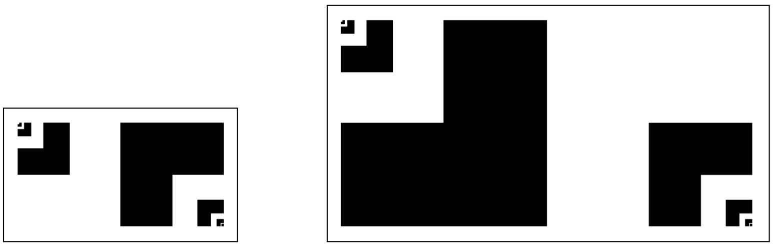


Figure 2: Matrices $M_{178}^{13,6}$ and $M_{178}^{15,7}$, where "178" stands for the elementary CA Rule 178.

Remark. *We can consider the more restricted* one-round communication complexity *measure. 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 [9], by simply counting the number of different rows or columns of a certain matrix we obtain the* exact *one-round communication complexity of the function. In our framework, the one round communication complexity of* $\textsc{Pred}_F|^i_n$ *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* $\textsc{Pred}_F|^i_n$, *becomes an easy task.*

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

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

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 CAs 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* $\mathrm{CC}\left(\textsc{Pred}_F\right) \leq g(n)/2$*.*

Following the work of M. Sablik [14], one can characterize the set of CAs having a bounded number of dependant cells (i.e, a bounded function $g(n)$): They are exactly these CAs which are equicontinuous in some direction (Theorem 4.3 of [14]). This set contains the nilpotent CAs (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*

$$\mathrm{CC}\left(\textsc{Pred}_F\right) \in O(1).$$

Another set of CAs with that property is the set of linear CAs. 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:

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

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

Proposition 3. *If* F *is a linear CA then* $\mathrm{CC}\left(\textsc{Pred}_F\right) \in O(1)$

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. □

3.2. Invasion

Let F be a CA and let u be a given word. Roughly, the problem $\textsc{Inv}_F^u$ is defined as follows: Given an input word w, we define the u-periodic configuration p_u on 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

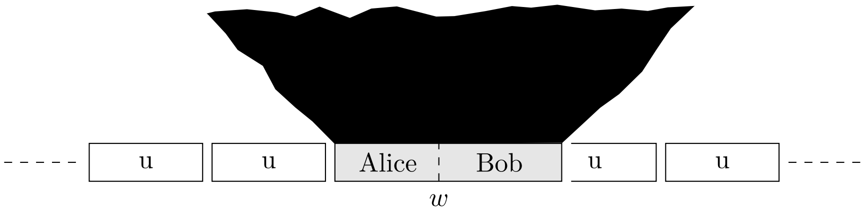


Figure 3: The INVASION problem

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 the classical algorithmic theory, undecidable.

Now we give formal definitions.

Definition 4. *Let $u = u_1 \dots u_l$ 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 $\big(F^t(p_u)\big)_t$ as the reference orbit;*
- *for each $x_1, \dots, x_n \in Q$, we define the configuration $p_u(x_1, \dots, x_n)$ obtained by modifying p_u as follows:*

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

- *for each t, we denote $\delta_l(t)$ and $\delta_r(t)$ the lefmost and rightmost differences between the t^{th} images of p_u and $p_u(x_1, \dots, x_n)$:*

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

- *then $\textsc{Inv}_F^u(x_1 \dots x_n)$ 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 $\textsc{Inv}_F^u$ defined as $\mathrm{CC}\left(\textsc{Inv}_F^u\right)$.

Some CAs have by nature a trivial invasion complexity because their dynamics consists in propagating errors systematically. This is the case of (positively) expansive CAs. 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* $\mathrm{CC}\left(\textsc{Inv}_F^u\right) = 1$.

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 [8], 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. □

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) perodic initial configuration. The formal definition follows.

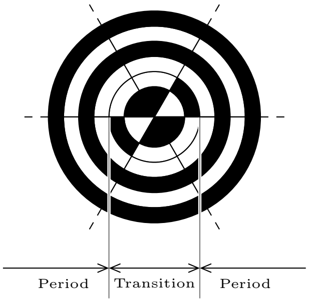


Figure 4: The Cycle problem on elementary CA Rule 33. Zeros are white, and ones are black. Initial configuration: 011011.

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:*

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

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 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:

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

Therefore, the following protocol of constant communication cost solves the problem $\textsc{Cycle}_F^k|_n$ on input c:

- Alice and Bob each know the evolution of a part of the configuration during the k first steps so they can communicate the list of time steps $t \leq k$ for whichthe part of the configuration they know becomes identical to the corresponding part of c.

- The part of the configuration that neither Alice nor Bob can know during the k first steps does not depend on n; therefore, after a constant amount of communication, Alice or Bob can have all the information to know this part during the k first time steps and produce the list of $t \leq k$ such that this part of the configuration becomes identical to the corresponding part of c;

- Finally, Alice or Bob can answer by simply checking whether the above lists of time steps have a non empty intersection.

□

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 maximal. Before giving precise definitions, recall that a CA is intrinsically universal if it is able to simulate any other CA. We are going to proceed 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 CAs 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 $\overline{\iota} \circ F = G \circ \overline{\iota}$, where $\overline{\iota} : Q_F^{\mathbb{Z}} \to Q_G^{\mathbb{Z}}$ denotes the uniform extension of ι.

A CA F simulates a CA G if some *rescaling* of F is a sub-automaton of some *rescaling* of G. The ingredients of the rescalings are simple: packing cells into blocs, 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:

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

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 $\mathbb{N}$:

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

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 focus on whether some functions belong or not to this class.*

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

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

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

Iterating: we have $\mathrm{CC}\left(\textsc{Pred}_{F^t}\right)(n) = \mathrm{CC}\left(\textsc{Pred}_F\right)(t \cdot n)$.

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

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

.

Reciprocally, one has for all n:

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

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

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

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*

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

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

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

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

The result follows by composition of the 3 properties above. □

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 $k' \geq k_0$;*
- $\mathrm{CC}\left(\textsc{Cycle}_F^k\right) \prec \mathrm{CC}\left(\textsc{Cycle}_G^{k'}\right)$.

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, $\mathrm{CC}\left(\textsc{Cycle}^k_F\right) \prec \mathrm{CC}\left(\textsc{Cycle}^k_G\right)$;
- for any k,
  - $\mathrm{CC}\left(\textsc{Cycle}^k_F\right) \prec \mathrm{CC}\left(\textsc{Cycle}^k_{F<m,1,0>}\right)$ and
  - $\mathrm{CC}\left(\textsc{Cycle}^k_{F<m,1,0>}\right) \prec \mathrm{CC}\left(\textsc{Cycle}^k_F\right)$;
- for any t and any k we have:
$$\mathrm{CC}\left(\textsc{Cycle}^k_{F<1,t,0>}\right) \prec \mathrm{CC}\left(\textsc{Cycle}^{kt}_F\right);$$
- for any t and any k such that $k \bmod t = 0$ we have:
$$\mathrm{CC}\left(\textsc{Cycle}^k_F\right) \prec \mathrm{CC}\left(\textsc{Cycle}^{k/t}_{F<1,t,0>}\right).$$

The proposition follows. □

4.2. Existence of CAs with maximal complexity

This section is devoted to the following existence result.

Proposition 9.

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

We now define the reversible CA of assertion 2 of Proposition 9 called 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.

- 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.

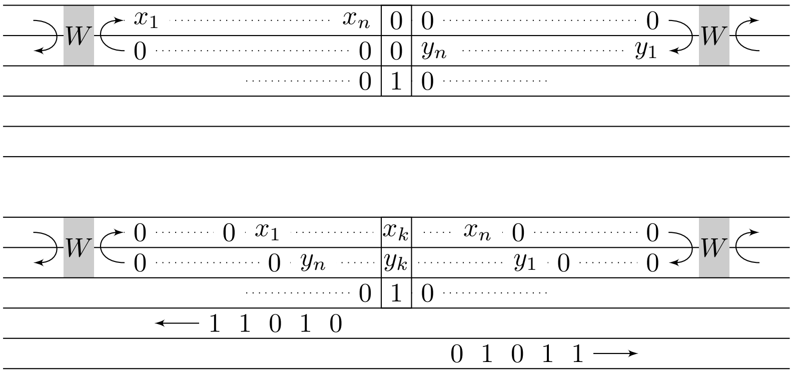


Figure 5: Above: initial configuration. Below: the configuration k steps later.

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.

- We first show that G defined above has the properties of assertion 2 of the proposition. First, it is reversible: flag and circulation layers are reversible by themselves and test layer is reversible when knowing flag and circulation layers.

  Now let q_0 be the state where flag layer is 0, circulation layer is $(0,0)$ and test layer is $(0,0)$. Consider input bits $x_1,\ldots,x_n$ on 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 $(x_i,0)$. Similarly 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 $\textsc{Inv}^{u}_{F^{2n+3}}$ 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 $\mathrm{CC}\left(\textsc{Inv}^{q_0}_{G}\right)\in\Omega(n)$.

- Assertion 1 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

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

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

  $$\textsc{Pred}_F|_n(u) = 1 \iff \sum x_i y_i \bmod 2 = 1.$$

  Proposition 1 implies that $\mathrm{CC}\left(\textsc{Pred}_F\right) \in \Omega(n)$.

- Assertion 3 of the proposition is proven by proposition 4.4.3.

□

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. $\mathrm{CC}\left(\textsc{Pred}_F\right) \in \Omega(n)$,
2. *there exists u s.t.* $\mathrm{CC}\left(\textsc{Inv}_F^u\right) \in \Omega(n)$,
3. *there exists k s.t.* $\mathrm{CC}\left(\textsc{Cycle}_F^k\right) \in \Omega(n)$.

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

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

A first application of this corollary to the complexity upper-bounds presented in Section 3 yields the following necessary conditions for universality. A first proof of these results appears in [15]. 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.*

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 CAs, $\mathcal{A}$ 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 launching signals invading the whole configuration, while keeping the prediction problem easy. For that purpose, we consider CA F, a cartesian product of two layers:

1. A layer performing a test for equality, as described below, and initialy containing a word over the alphabet $\{\overrightarrow{0}, \overrightarrow{1}, \overleftarrow{0}, \overleftarrow{1}, \top, \emptyset, K\}$. On figure 6, this layer is represented in black.
2. A layer with an automaton invading the configuration from a seed. We need five states on this layer: $\{s, \emptyset, \rightarrow, \leftarrow, K\}$. We describe the rule below. On figure 6, this layer appears in blue.

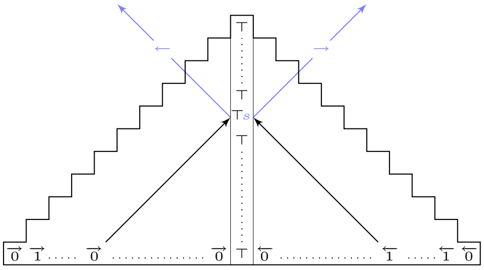

←
→
⊤
⊤
⊤s
⊤
$\overrightarrow{0}$ $\overrightarrow{1}$ ..... $\overrightarrow{0}$ .... $\overrightarrow{0}$ ⊤ $\overleftarrow{0}$ .... $\overleftarrow{1}$ ..... $\overleftarrow{1}$ $\overleftarrow{0}$

Figure 6: A CA easy for Pred and hard for Inv

The simplest way to embed an equality test in CAs is to ask Alice and Bob whether the input is a palindrom. There is an easy *fooling set* (see definition 1 or [9]) to ensure that this problem has a high deterministic communication complexity. The K state is spreading on both layers, i.e. if it appears somewhere in one layer, then it spreads everywhere on both layers. We use it to detect ill-formed configurations where a $\overrightarrow{x}$ state is next to a $\overleftarrow{x}$, or there are several s states on the second layer.

On configurations not involving the K state, the local rule is the right shift on words containing only $\overleftarrow{x}$ states, a $\top$ or $\emptyset$ state on their right, and the left shift on words containing only $\overrightarrow{x}$ states, a $\top$ or $\emptyset$ state on their left, where x ranges over $\{0, 1\}$. States in $\{\top, \emptyset\}$ remain unchanged, and all other transitions yield state K.

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

On the second layer, the K state is spreading, and for all input words not containing K, the local rule is the following:

$$\begin{array}{rcl} f(s, \emptyset, \emptyset) & = & \rightarrow \\ f(\rightarrow, \emptyset, \emptyset) & = & \rightarrow \\ f(\emptyset, \emptyset, s) & = & \leftarrow \\ f(\emptyset, \emptyset, \leftarrow) & = & \leftarrow \\ \text{else } f(x, y, z) & = & \emptyset \end{array}$$

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

- $\mathrm{CC}\,(\textsc{Pred}_F) \in O(1)$,
- *there is u such that* $\mathrm{CC}\,(\textsc{Inv}_F^u) \in \Omega(n)$.

Proof. A protocol for Pred needs to predict the content of both layers:

- 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, the result is a s state if and only if the leftmost state of Alice's differs from the rightmost state of Bob's, and the central cell is a $\top$ state. Else, it can be a $\rightarrow$ state (the $\leftarrow$ case is symmetric) if there was an s state on the left edge of the computation triangle: the protocol for deciding this is the same as before, in a smaller triangle (see figure 7). But this has only to be checked for the leftmost $\top$ state of Alice's, since even if other signals are to be generated afterwards, they do not appear on the top of the bigger triangle.

  In all other cases, the second component is empty.

  In all cases, predicting the final state of this layer requires a constant amount of communication.

Now we need to find hard instances for the Inv problem: With a repeated word containing only $\emptyset$ states on both layers, on 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. Proposition 1 is concluded. □

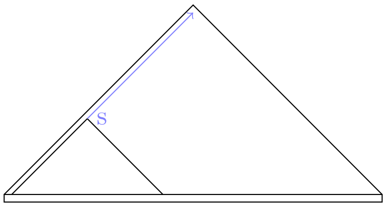


Figure 7: Our protocol.

4.4.2. A CA easy for Inv *and hard for* Pred

There is a natural example described in section 6.1. It remains to show that the deterministic (possibly with several rounds) communication complexity of the Pred problem is in $\Omega(\log n)$. To show this, we construct a fooling set S_n (see Definition 1 or [9]):

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

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 $\textsc{Pred}_{F_{218}}$ 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 8. Thus, since $|S_n| = n+1$, we deduce that a deterministic protocol solving $\textsc{Pred}_{F^n_{218}}$ can not take less than $\log(n+1)$ steps:

$$\mathrm{CC}\left(\textsc{Pred}_{F^n_{218}}\right) \in \Omega(\log(n))$$

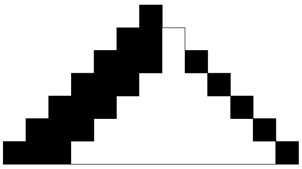


Figure 8: A configuration of the fooling set S_n for rule 218

However, from Section 6.1, the Inv problem can be solved within constant communication.

4.4.3. A CA easy for Inv *and hard for* Cycle

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 and a uniform periodic configuration (i.e. 1-periodic).

We use three layers in this construction: One always shifting right, one other always shifting left, and the third either empty or performing a test about the two other layers. More precisely, we use the layout of Figure 9.

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 9), and a third

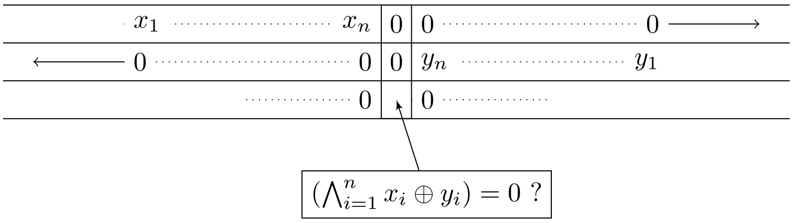


Figure 9: An automaton with a hard Cycle problem, and an easy Inv.

layer everywhere empty, except for a central "test" state, actually performing the tests. As stated in Figure 9, the performed test is $\bigwedge_{i=1}^n x_i \oplus y_i = 0$, with $\oplus$ denoting the binary "xor" operation (the test cell at time $t+1$ takes value $T_t \wedge (x \oplus y)$ where T_t is its value at time t and x and y are the values on the two first layers). While the test value is 1, the tests go on. Since the tape is cyclic, if $\bigwedge_{i=1}^n x_i \oplus y_i = 1$, then the test goes on forever producing a (temporal) cycle of length $\Omega(n)$ (because in such a case, we have at least one x_i or one y_i which is 1). 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 . In the sequel, we call this CA F.

Proposition 11. *The CA F described above is such that:*

- $\forall k, \mathrm{CC}\left(\textsc{Cycle}_F^k\right) \in \Omega(n)$.
- $\forall u, \mathrm{CC}\left(\textsc{Inv}_F^u\right) \in O(1)$,

Proof. The first assertion follows from the discussion above.

It remains to show that $\textsc{Inv}_F^u$ can be solved within constant communication. Let u be any word over the alphabet for F. First if the orbit of p_u contains a spreading state, since p_u is periodic, then after a constant number of steps, nothing can happen, thus the input word is ignored and no invasion can occur. In all other cases, we note the input word w as (x, y) where $x = \pi_1(w)$ is the word appearing on the first component and $y = \pi_2(w)$ the word appearing on the second. The protocol consists in discussing whether some components are equal between p_u and $p_u(w)$. Each of the following case can be tested with constant communications:

- If $x \neq \pi_1(p_u[1..|x|])$, and $y \neq \pi_2(p_u[1..|y|])$, then there is always invasion: if a spreading state is ever generated, it invades the configuration (by hypothesis, there is no spreading state in the orbit of p_u), else, x and y are continually shifted, so there is also invasion.
- If $p_u((x, y)) = p_u$ then there is no invasion by Definition 4.
- Otherwise we have necessarily $x \neq \pi_1(p_u[1..|x|])$, and $y = \pi_2(p_u[1..|y|])$ (or the same inverting $=$ and $\neq$, but the reasoning is similar). In this case, the discussion is the following:
  - if the position of tests in the third component is the same in p_u and $p_u(w)$ (which can be decided with constant communication) then Alice and Bob each know whether their part of the first component will

ever provoke a test failure and generate a spreading state; therefore they can decide invasion: there is invasion in case of a test failure, and there is no invasion else because the differences between p_u and $p_u(w)$ will only be shifted and will never spread to an infinite width;

- if the position of the tests are different, then there is always invasion: either because some test failure generates a spreading state, or because the differences between p_u and $p_u(w)$ in the first component are shifted towards infinity, but a difference in the third component (a test position) stay at the same place indefinitely.

□

4.4.4. *A CA easy for* Pred*, and hard for* Cycle

We can use quite the same construction as in Subsection 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 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 10). Therefore, the period of the configuration is $\Omega(n)$ if and only if an error occurs.

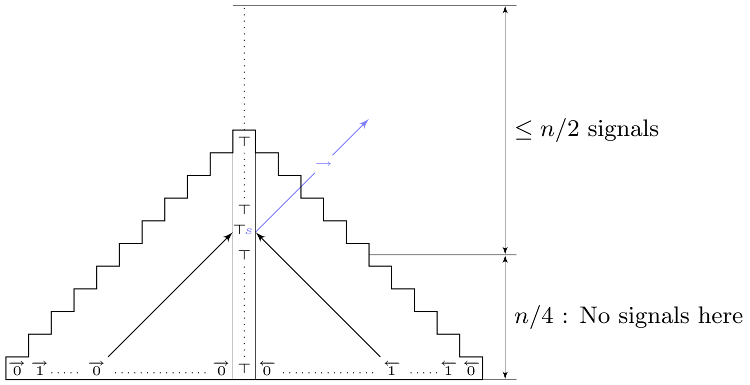

$\leq n/2$ signals
$n/4$: No signals here
$\top$
$\top$
$\top_s$
$\top$
$\rightarrow$
$\overrightarrow{0}$ $\overrightarrow{1}$ ..... $\overrightarrow{0}$ .... $\overrightarrow{0}$ $\top$ $\overleftarrow{0}$ .... $\overleftarrow{1}$ ..... $\overleftarrow{1}$ $\overleftarrow{0}$

Figure 10: A CAeasy for Pred and hard for Cycle.

4.4.5. *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 $\Omega(\log n)$.

4.4.6. *A CA easy for* Cycle *and hard for* Inv

We can reuse the construction of Subsection 4.4.1, and use the cyclicity of the configuration to find an easy protocol: For all ill-formed configurations, the

CA is periodic with state K spreading to all the configuration. For any other configuration, even if there is a seed somewhere on the second layer, the signals it sends cross after some time, resulting in a $\emptyset$ state on this layer after at most a $O(n)$ steps. Moreover, the $\top$ states do not move, so the configuration becomes 1-periodic after all the tests are performed: There may be a column of $\top$ states, and $\emptyset$ states everywhere else.

Thus, since this CA is always 1-periodic, the Cycle problem can be decided with no communication.

5. Intrinsic universality: Ruling out complex CAs

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 measure.

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 CAs from being intrinsically universal.

5.1. Prediction

T. Neary and D. Woods proved “the P-completeness of Rule 110” [12]. In our language, they proved that the problem $\textsc{Pred}_{F_{110}}$ is P-complete. A very natural question arises: What do classical algorithmic properties of CAs, such as P-completeness, imply on the its communication complexity counterpart?

Therefore, the communication complexity approach appears to be a promising tool for ruling out CAs from being intrinsically universal. More precisely, despite the fact that the decision version of a canonical problem is hard (P-complete, undecidable) its corresponding communication complexity might be rather low.

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

$$\mathrm{CC}\left(\textsc{Pred}_F\right) \in O(n^{1/k})$$

and $\textsc{Pred}_F$ *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 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 heads 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 11 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.

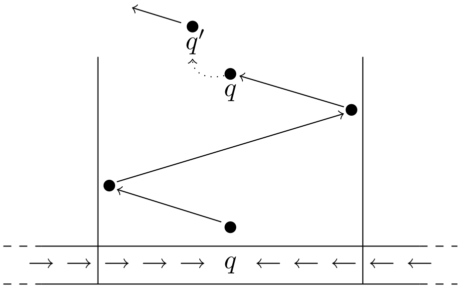

q'
q
$\rightarrow\ \rightarrow\ \rightarrow\ \rightarrow\ \rightarrow\ q\ \leftarrow\ \leftarrow\ \leftarrow\ \leftarrow\ \leftarrow$

Figure 11: A CA for which Pred is P-complete.

□

Remark. *A result by Hromkovic (see [7]) says 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 computation" power of Alice and Bob, it can be decided within very little communication.

Proposition 13.

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

Proof.

1. Let F and u be fixed and consider the problem $\textsc{Inv}_F^u$. Given an input $x_1, \ldots, x_n$, 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 convinient choice for β. □

2. We build a CA F that simulates a 2-counter machine [11]. 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 → and ← 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

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

   A valid configuration is a configuration everywhere equal to $\emptyset$ except on finite coding segments which have the folloing form (see figure 12):

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

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

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

- 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 repeatidely from one extremity of the segment to the other and extend the segment at each passage 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 passage 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 to 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); such a 0 must exist 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 $\textsc{Inv}_F^u$ (halt $\iff$ no invasion). Therefore, by a suitable choice of the 2-counter machine used to construct F, we have that $\textsc{Inv}_F^u$ is Π_1^0-complete.

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

- 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.

□

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 $\textsc{Cycle}_F^k$ is PSPACE-complete for some $k \in \mathbb{N}$, and $\mathrm{CC}\left(\textsc{Cycle}_F^k\right) \in o(n)$.

Proposition 14. 1. *For any CA F and any $k \geq 1$, $\textsc{Cycle}_F^k \in$ PSPACE.*
2. *Their exist F and k such that $\textsc{Cycle}_F^k$ 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.

   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 13. It first directly translates x into states of F, then computes $P(|x|)$ and outputs $P(x)$ blank states.

□

6. Intrinsic universality: Ruling out concrete elementary CAs

6.1. CA Rule 218

The local function $f_{218} : \{0,1\}^3 \to \{0,1\}$ of CA Rule 218 is defined in Figure 14(a).

From the result of [5] we already knew that $\mathrm{CC}\left(\textsc{Pred}_{F_{218}}\right) \in O(\log(n))$. 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:

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

□

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^n1 with $n \geq 2$ is $10^{n-2}1$. □

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

Proof. First, if the configuration p_u is non-additive then, by Lemma 3, at some time t a wall appears periodically in $F^t_{218}(p_u)$. 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. □

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

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).

Here appears clearly how powerful the invasion approach is (as a tool for proving non-universality). Finding an upper bound (a protocol) for CC $(\textsc{Pred}_{F_{94}})$ 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* $\big((00)^+(11)^+\big)^*$ *(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. □

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 $t \geq 0$).*

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 $n \geq 2$. □

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. □

From the 2 lemmas above, we get that

Proposition 16. *For any u we have* CC $\left(\textsc{Inv}^u_{F_{94}}\right) \in O(\log(n))$.

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 exchanging a 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 it must also contain 101. Consider the leftmost and the rightmost occurrences of such kind of words. Since walls appear above the middle of that 2 occurrences after a time equal to their half-lengths (Lemma 5), the fact there is invasion or not does not

depend on what is beetween that two occurrences. It take $O(\log(n))$ bits of communications for Alice to know the positions of that 2 occurrences and the exact words present at those 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 1 difference with p_u in those additive part, then there is invasion. If not, then Alice has all information to decide invasion. Deciding in which of the two cases we are can be done exchanging a constant number of bits.

□

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 $f_{33} : \{0,1\}^3 \to \{0,1\}$ of CA Rule 33 is defined in Figure 16(a).

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

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 $xyz \in \{000 \dots 111\}$, and $u \in A_0 \cap A_{00}$, we observe that:

$$\forall u \in A_0 \cap A_{00}, |u| = 7, (F_{33})^2(u_1 \dots u_7) = u_3u_4u_5$$

□

Lemma 8. *All (cyclic) configurations of length n, different from* $(01)^{\lfloor n/2 \rfloor}$*, do not contain isolated* 0*s 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 $(01)^{\lfloor n/2 \rfloor}$.

□

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

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

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

Proposition 17. *A deterministic protocol for predicting CA Rule 33 can not be in* $o(\log n)$.

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

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

It can be easily verified that:

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

Since $|S_n| = \left\lfloor \frac{n}{2} \right\rfloor$, we conclude that a deterministic protocol for CA Rule 33 needs at least $\Omega(\log_2 n)$ bits of communication.

□

7. Conclusion

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

- It seems that more can be said about the communication complexity problems for the class of surjective CAs 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 CAs 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 CAs, 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 CAs (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.

References

[1] Laurent Boyer and Guillaume Theyssier. On local symmetries and universality in cellular automata. In *Proceedings of the 26th International Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 195–206, 2009.

[2] Matthew Cook. Universality in elementary cellular automata. *Complex Systems*, 15:1–40, 2004.

[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. *Theoretical Computer Science*, 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 Seok-Hee Hong and Hiroshi Nagamochi, editors, *Proceedings of the 19th International Symposium on Algorithms and Complexity (ISAAC 2008), Lecture Notes in Computer Science 5369*, volume 2380 of *Lecture Notes in Computer Science*, pages 71–94. Springer, 2008.

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

[7] Juraj Hromkovic and Georg Schnitger. Communication complexity and sequential compuation. In *MFCS '97: Proceedings of the 22nd International Symposium on Mathematical Foundations of Computer Science*, pages 71–84, London, UK, 1997. Springer-Verlag.

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

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

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

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

[12] Turlough Neary and Damien Woods. P-completeness of cellular automaton Rule 110. In *In International Colloquium on Automata Languages and Programming (ICALP), volume 4051 of LNCS*, pages 132–143. Springer, 2006.

[13] Nicolas Ollinger. Universalities in cellular automata: a (short) survey. In B. Durand, editor, *Symposium on Cellular Automata Journées Automates Cellulaires (JAC'08)*, pages 102–118. MCCME Publishing House, Moscow, 2008.

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

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

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

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

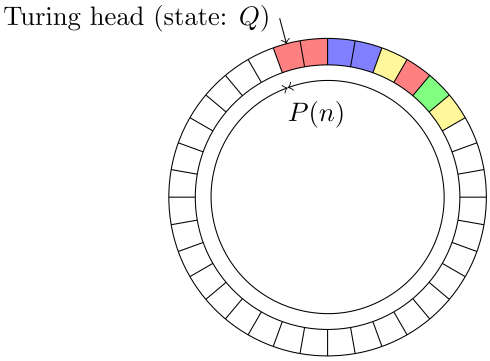
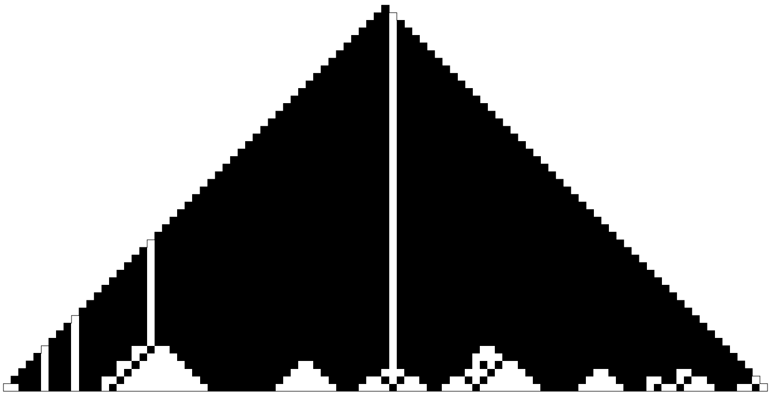


Figure 13: The output of the transducer used in Proposition 14.

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}.

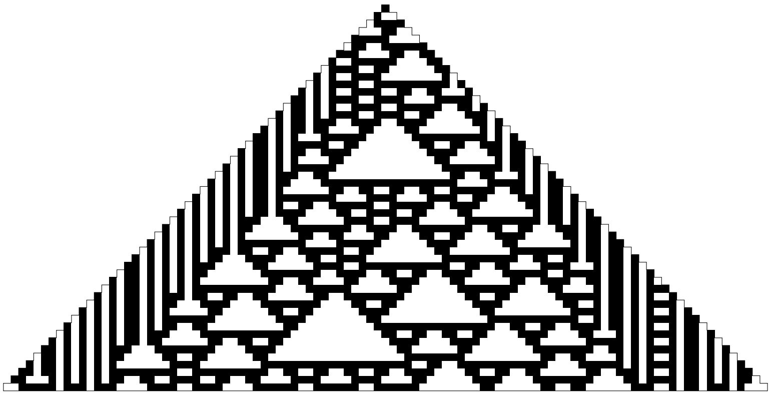


(b) Example of a space-time diagram for CA Rule 218.

Figure 14: CA Rule 218.

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

(a) f_{94}.

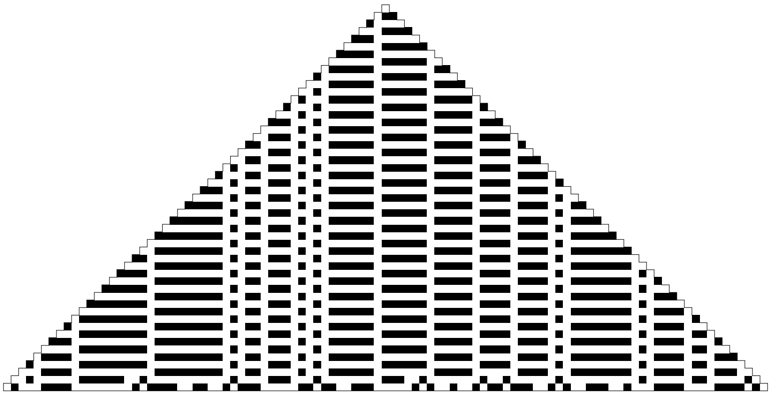


(b) Example of a space-time diagram for CA Rule 94.

Figure 15: CA Rule 94.

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

(a) f_{33}.



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

Figure 16: CA Rule 33.