Understanding a Non-Trivial Cellular Automaton by Finding its Simplest Underlying Communication Protocol *

Eric Goles[1,2], Cedric Little[1], and Ivan Rapaport[2,3]

[1] Facultad de Ciencias y Tecnologías, Universidad Adolfo Ibáñez, Chile
[2] Centro de Modelamiento Matemático (UMI 2807 CNRS), Universidad de Chile
[3] Departamento de Ingeniería Matemática, Universidad de Chile

Abstract. In the present work we find a *non-trivial* communication protocol describing the dynamics of an elementary CA, and we prove that there are *no simpler* descriptions (protocols) for such CA. This is, to our knowledge, the first time such a result is obtained in the study of CAs. More precisely, we divide the cells of Rule 218 into two groups and we describe (and therefore understand) its global dynamics by finding a protocol taking place between these two parts. We assume that $x \in \{0,1\}^n$ is given to Alice while $y \in \{0,1\}^n$ is given to Bob. Let us call $z(x,y) \in \{0,1\}$ the result of the dynamical interaction between the cells. We exhibit a protocol where Alice, instead of the n bits of x, sends $2\lceil \log(n) \rceil + 1$ bits to Bob allowing him to compute $z(x,y)$. Roughly, she sends 2 particular positions of her string x. By proving that any one-round protocol computing $z(x,y)$ must exchange at least $2\lceil \log(n) \rceil - 5$ bits, the optimality of our construction (up to a constant) is concluded.

1 Introduction

The process of understanding and classifying cellular automata (CAs) has been carried out mainly by researchers belonging to the dynamical systems community [2, 9, 14]. This interest can be explained on one hand by the simple fact that CAs *are* discrete dynamical systems and, on the other hand, by the impact of Wolfram's classification [21], which is an "empirical categorization of space-time pattterns into four classes loosely based on an analogy with those found in continous state dynamical systems"; nevertheless, this classification "has resisted numerous attempts at formalizations" [7].

We claim that CAs are extremely complex (highly non linear) objects and therefore the language of computer science appears to be particularly suitable for studying them. More precisely, our approach is to divide the cells into (two) groups in order to describe the dynamics by finding simple communication protocols taking place between these parts.

* Partially supported by Programs Conicyt "Anillo en Redes", Fondap, Basal-CMM, Fondecyt 1070022 and Instituto Milenio ICDB.

Obviously, this is not the first time CAs are analyzed from a (theoretical) computer science point of view. The algorithmic approach has always been present. In fact the model itself was invented in the 1950's as a tool to study self-reproduction [16]. And more recently researchers have tackled different algorithmic problems ranging from the intrinsic universality and the complexity of predicting [4, 11, 12, 15, 17, 20] to the decidability/complexity of different dynamical systems properties [1, 3, 6, 8]. But the present work is, to our knowledge, the first one where non-trivial protocols are discovered in the dynamics itself (in a previous paper the connection between CAs and communication complexity began to be explored [5]; nevertheless, in that work, instead of understanding the CAs behavior, the main interest was to give a formal classification; in fact, proofs were given just for simple cases).

1.1 Basics. An (elementary) CA is defined by a local function $f : \{0,1\}^3 \to \{0,1\}$, which maps the state of a cell and its two immediate neighbors to a new cell state. There are $2^{2^3} = 256$ CAs and each of them is identified with its Wolfram number $\omega = \sum_{a,b,c\in\{0,1\}} 2^{4a+2b+c} f(a,b,c)$ (see [21, 22]). Sometimes, instead of expliciting function f, we refer to f_ω. The dynamics is defined in the one-dimensional cellspace. Following the CAs paradigm, all the cells change their states synchronously according to f. This endows the line of cells with a global dynamics whose links with the local function are still to be understood. After n time steps the value of a cell depends on its own initial state together with the initial states of the n immediate left and n immediate right neighbor cells. More precisely, we define the n-th iteration $f^n : \{0,1\}^{2n+1} \to \{0,1\}$ recursively: $f^1(z_{-1}, z_0, z_1) = f(z_{-1}, z_0, z_1)$ and, for $n \geq 2$,

$$f^n(z_{-n} \dots z_1, z_0, z_1 \dots z_n) = f^{n-1}(f(z_{-n}, z_{-n+1}, z_{-n+2}) \dots f(z_{n-2}, z_{n-1}, z_n)).$$

This work is motivated by the following idea: if we were capable of giving a simple description of f^n (for arbitrary n) then we would have understood the behavior of the corresponding CA.

1.2 Representation. The first step is to represent f^n as two families of 0-1 matrices depending on whether the central cell begins in state $c = 0$ or $c = 1$. More precisely, the square matrices $M_f^{c,n}$ of size 2^n are defined as follows (see Figure 1).

$$M_f^{c,n}(x,y) := f^n(x,c,y) \text{ with } x = x_n \dots x_1 \text{ and } y = y_1 \dots y_n \text{ in } \{0,1\}^n.$$

Note that the first matrix of each family, standing for $n = 1$, completely defines the local function. One can think of these matrices as seeds for the families. We should emphasize also that the space-time diagram shows the evolution of only a *single configuration*, while the matrix covers *all configurations*.

1.3 Interpretation. This step is obviously the most difficult. Here we try to prove and interpret the behavior of $M_f^{c,n}$ for arbitrary values of n. Fortunately, these 0-1 matrices reveal themselves to be a striking representation. For instance,

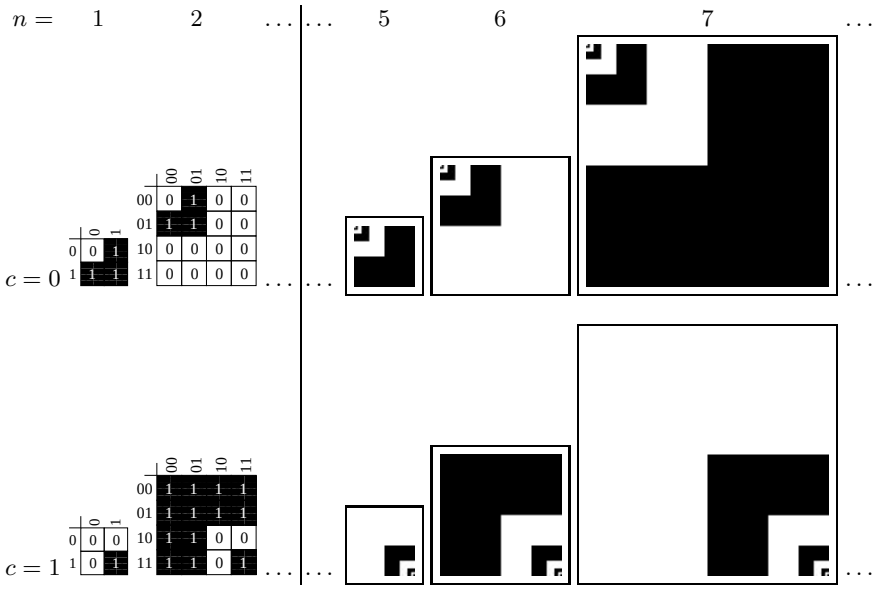

Fig. 1. The two families of binary matrices $M^{c,n}_{f_{178}}$ of Wolfram Rule 178.

let us consider Rule 105. In Figure 2 we show on the left the space-time diagram of Rule 105 for some initial configuration, and on the right the matrix $M^{0,6}_{f_{105}}$. In contrast with the space-time diagram, the matrix looks simple. In fact, as we are going to see later, the simplicity of a matrix $M^{c,n}_f$ is related to the simplicity of the communication protocol that computes f^n. Therefore, assuming that $x = x_n \dots x_1 \in \{0,1\}^n$ is given to one party (say Alice) and that $y = y_1 \dots y_n \in \{0,1\}^n$ is given to another party (say Bob), we are going to look for the simplest communication protocols that compute both $f^n(x,0,y)$ and $f^n(x,1,y)$.

1.4 Our contribution. Let $d(M)$ be the number of different rows of a matrix M. In [5] the only CAs we managed to explain were those we called *bounded* (where $d(M^{c,n}_f)$ was constant) and *linear* (where $d(M^{c,n}_f)$ grew as $\Theta(n)$). All the other CAs were grouped together using a mainly experimental criterion. We conjectured the existence of *polynomial* and *exponential* classes. In the present work we prove the existence of a CA for which $d(M^{c,n}_f)$ grows as $\Theta(n^2)$.

Linear and bounded rules are easy to explain in terms of communication protocols. This is the case of Rule 178 of Figure 1 (this particular rule has just been studied by D. Regnault [18] using percolation theory and considering asynchronicity; we belive that the linearity of the rule and the fact that it is amenable to other types analysis is not a coincidence).

This paper shows that as soon as we move up in the hierarchy the underlying protocols become rather sophisticated. In fact, for the CA we treat here (Rule 218), we prove that if $c = 0$ then Alice needs to send 2 positions of her string

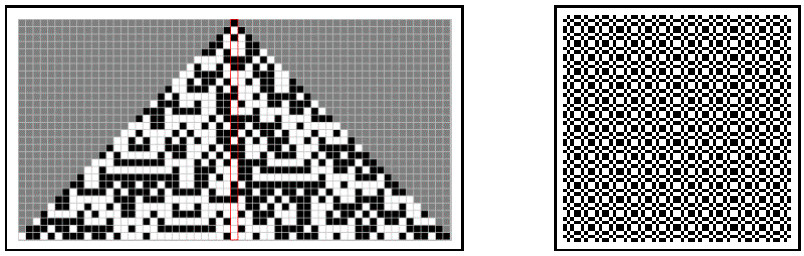

Fig. 2. A space time diagram for Rule 105 (left) and matrix $M_{f_{105}}^{0,6}$ (right). In the diagram every row is a configuration and time goes upward; grey cells represent states which are undetermined from the bottom (initial) configuration.

($2\log(n)$ bits). The quantitative relation between *the one-round communication complexity* and *the number of different rows* will be explained later. But roughly, the first is the logarithm of the second. Therefore, sending $2\log(n)$ bits is equivalent to having $\Theta(n^2)$ different rows. The difference between sending 1 position ($\Theta(n)$ behavior) and 2 positions ($\Theta(n^2)$ behavior) is huge. The reader can verify this by comparing the cases $c = 0$ and $c = 1$.

We think Rule 218 is one of the few CAs for which a non-trivial behavior can be proven. Experimentally, we do not find many candidates in a class $\Theta(n^k)$ with $k \geq 2$. This could imply that there are no other CAs with simple descriptions (shortcuts). We should also point out that, if more than 2 states were allowed, we could build CAs with arbitrary complexity. In fact, in [5] it is shown how to construct a 3 state CA exhibiting a $\Theta(n^3)$ behavior. But in the present work we are dealing with the *inverse problem*.

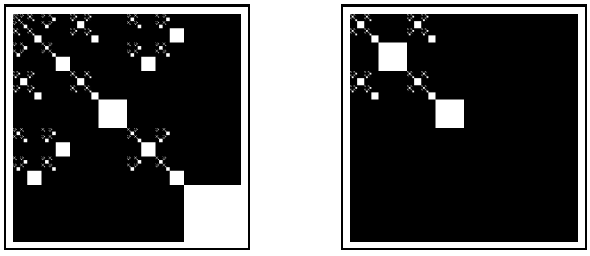

Fig. 3. $M_{f_{218}}^{0,9}$ (left) and $M_{f_{218}}^{1,9}$ (right).

1.5 Rule 218. The local function of CA Rule 218 is the following:

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

Its global dynamics is represented by the two matrices of Figure 3 and by the space-time diagrams of Figure 4 (Rule 218 and Rule 164 are the same; 0s behave as 1s and viceversa). We encountered Rule 218 when trying to find a (kind of) double-quiescent palindrome-recognizer. Despite the fact that it belongs to class 2 (according to Wolfram's classification), it mimics Rule 90 (class 3) for very particular initial configurations.

Authors in [13] were surprised when they found, "unexpectedly", that the rule exhibited $1/f^{\alpha}$ spectra. Rule 218 has also been proposed as a symmetric cipher [19]. Nevertheless, it should be clear that the most relevant aspect of Rule 218 is its behavior in this communication context. More precisely, Rule 218 seems to be one of the most complex CAs for which a reasonable protocol can be found.

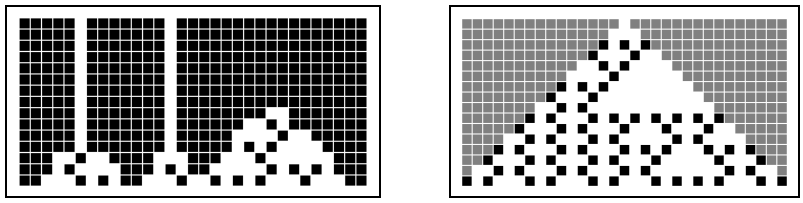

Fig. 4. Two space-time diagram for Rule 218. Every row is a configuration and time goes upward. Grey cells represent states which are undetermined from the bottom (initial) configuration.

2 Two-party protocols

The communication complexity theory studies the information exchange required by different actors to accomplish a common computation when the data is initially distributed among them. To tackle that kind of questions, A.C. Yao [23] suggested the two-party model: two persons, say Alice and Bob, are asked to compute together $f(x, y)$, where Alice knows x only and Bob knows y only (x and y belonging to finite sets). Moreover, they are asked to proceed in such a way that the cost –the total number of exchanged bits– is minimal in the worst case. Different restrictions on the communication protocol lead to different communication complexity measures. Whereas most studies concern the many-round communication complexity, we focus only on the one-round.

Definition 1 (One-round communication complexity). *A protocol $\mathcal{P}$ is an AB-one-round f-protocol if only Alice is allowed to send information to Bob, and Bob is able to compute the function solely on its input and the received information. The cost of the protocol $c_{AB}(\mathcal{P})$ is the (worst case) number of bits Alice needs to send. Finally, the AB-one-round communication complexity of a*

function f is $c_{AB}(f) = c_{AB}(\mathcal{P}^*)$, where $\mathcal{P}^*$ is an AB-one-round f-protocol of minimum cost. The BA-one-round communication complexity is defined in the same way.

The following fact throws light on the interest of the one-round communication complexity theory for our purpose: we can infer the exact cost of the optimal AB-one-round protocol by just counting the number of different rows in the matrix.

Fact 1 ([10]) *Let f be a binary function of $2n$ variables and $M_f \in \{0,1\}^{2^n \times 2^n}$ its matrix representation, defined by $M_f(x,y) = f(xy)$ for $x, y \in \{0,1\}^n$. Let $d(M_f)$ be the number of different rows in M_f. We have $c_{AB}(f) = \lceil \log(d(M_f)) \rceil$.*

Example 1. Consider Rule 90, which is defined as follows: $f(a,b,c) = a + c$ (the sum is mod 2). This is an additive rule and it satisfies the superposition principle. More precisely, for every $x_n \dots x_1 \in \{0,1\}^n$, $\tilde{x}_n \dots \tilde{x}_1 \in \{0,1\}^n$, $y_1 \dots y_n \in \{0,1\}^n$, $\tilde{y}_1 \dots \tilde{y}_n \in \{0,1\}^n$, $c, \tilde{c} \in \{0,1\}$:

$$f^n(x_n \dots x_1, c, y_1 \dots y_n) + f^n(\tilde{x}_n \dots \tilde{x}_1, \tilde{c}, \tilde{y}_1 \dots \tilde{y}_n) = f^n(x_n + \tilde{x}_n \dots, c + \tilde{c}, \dots y_n + \tilde{y}_n).$$

Therefore, there is a simple one-round communication protocol. Alice sends one bit b to Bob. The bit is $b = f^n(x_n \dots x_1, c, 0 \dots 0)$. Then Bob outputs $b + f^n(0 \dots 0, 0, y_1 \dots y_n)$. The same superposition principle holds for Rule 105 of Figure 2. This simple protocol (together with Fact 1) explains why the number of different rows is just 2.

3 The protocols of Rule 218

Since Rule 218 is symmetric we are going to assume, w.l.g., that Alice is the party that sends the information. Moreover, we are going to refer simply to one-round protocols or one-round communication complexity (because the AB and BA settings are in this case equivalent). We denote f_{218} simply by f.

Notice that we can easily extend the notion of t iterations to blocks of size bigger than $2t+1$. In fact, for every $m \geq 2t+1$ and every finite configuration $z = z_1 \dots z_m \in \{0,1\}^m$ we define $f^0(z) = z$,

$$f^1(z) = (f(z_1, z_2, z_3), \dots, f(z_{m-2}, z_{m-1}, z_m)) \in \{0,1\}^{m-2}$$

and, recursively, $f^t(z) = f^{t-1}(f(z)) \in \{0,1\}^{m-2t}$.

Let $c \in \{0,1\}$. Let $x, y \in \{0,1\}^n$. From now on in this section, in order to simplify the notation, we are always assuming that these arbitrary values (i.e., n, c, x, y) have already been fixed.

Definition 2. *We say that a word in $\{0,1\}^*$ is additive if the 1s are isolated and every consecutive couple of 1s is separated by an odd number of 0s.*

Lemma 1. *If $xcy \in \{0,1\}^{2n+1}$ is additive, then $f^n(x,c,y) = f^n(x,c,0^n) + f^n(0^n,0,y)$.*

Proof. Rule 218 is "almost" the same as Rule 90 which is defined as follows: $(b_{-1}, b_0, b_1) \rightarrow b_{-1} + b_1$. The only case where the two rules differ is when $b_{-1}b_0b_1 = 111$. But for additive configurations the pattern 111 never appears and therefore its dynamics corresponds to the one of Rule 90. This rule is additive and therefore the superposition principle applies. □

Notation 1 *Let α be the maximum index i for which $x_i \dots x_1 c$ is additive. Let β be the maximum index j for which $cy_1 \dots y_j$ is additive. Let $x' = x_\alpha \dots x_1 \in \{0,1\}^\alpha$ and $y' = y_1 \dots y_\beta \in \{0,1\}^\beta$.*

Notation 2 *Let l be the minimum index i for which $x_i = 1$. If such index does not exist we define $l = 0$. Let r be the minimum index j for which $y_j = 1$. If such index does not exist we define $r = 0$.*

3.1 The lemmas

In this subsection we present all the lemmas we need in order to conclude the correctness of the protocols. These protocols are going to be presented in the next subsection. One could therefore begin by reading subsection 3.2 and check the lemmas later.

Lemma 2. $f^n(x, c, y) = f^n(1^{n-\alpha}x', c, y) = f^n(x, c, y'1^{n-\beta}) = f^n(1^{n-\alpha}x', c, y'1^{n-\beta})$.

Proof. By symmetry it is clear that it is enough to prove $f^n(x, c, y) = f^n(1^{n-\alpha}x', c, y)$. If $\alpha = n$ then it is direct. If $\alpha < n$ then there is a non-negative integer s such that $x_{\alpha+1} \dots x_{\alpha-2s} = 10^{2s}1$ (notice that s could be 0). It follows that $f^s(10^{2s}1) = 11$. Notice that a word 11 acts as a wall through which information does not flow. In fact, for all $b \in \{0,1\}$, $f(b,1,1) = f(1,1,b) = 1$. Therefore we conclude that the result is independent of the information to the left of position $\alpha + 1$ and we can assume, w.l.g, that $x_n \dots x_{\alpha+1} = 1^{n-\alpha}$. □

Definition 3. *A string z is called left additive if it satisfies one of the two following conditions: either (i) $z = 0 \dots 0$, or (ii) z is additive while $1z$ is not. For the right additivity definition we replace $1z$ by $z1$.*

Lemma 3. *Let $1 \leq s \leq n$. Let $z \in \{0,1\}^{2n+1-s}$. If z is left additive then $f(1^s z) = 1^s u$ with $u \in \{0,1\}^{2n-1-s}$ being left additive. If z is right additive then $f(z1^s) = u1^s$ with $u \in \{0,1\}^{2n-1-s}$ being right additive.*

Proof. We will prove the left additivity case (the right case is analogous). First we need to prove that the block of 1s moves to the right (see Figure 5 a). More precisely, that $f(1, z_1, z_2) = 1$. We know that $1z_1z_2 \neq 101$ because in that case $1z$ would have been additive. Therefore $f(1, z_1, z_2) = 1$. On the other hand, since $f(z) = u$, we know that u is additive. Now we need to prove that $u = 0 \dots 0$ or that $1u$ is not additive. Let us analyze two cases.

Case $z_1 = 0$. If $z = 0 \dots 0$ then $u = 0 \dots 0$. The other possibility is that z_1 belongs to an even length block of 0s (bounded by two 1s). If the length is 2

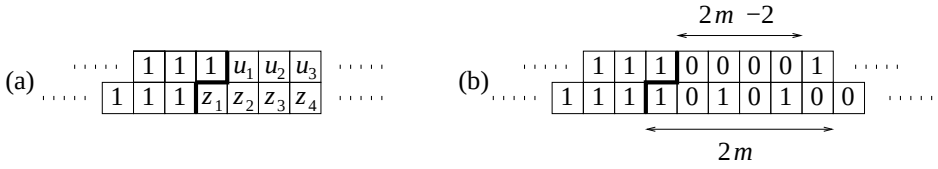

Fig. 5. a.- $f(1^s z) = 1^s u$. **b.-** $0^{2m-2}1$ is a prefix of u.

then $u_1 = f(z_1, z_2, z_3) = f(0, 0, 1) = 1$ and therefore $1u$ is not additive. If the length is even but bigger than two then the block shrinks in its two extremities and it remains even. Therefore, $1u$ also is not additive.

Case $z_1 = 1$**.** By the additivity of z we know that $z_2 = 0$. If $z_3 = 0$ then $u_1 = 1$ and therefore $1u$ is not additive. Hence let us assume $z_3 = 1$ (see Figure 5 b). Since $z_1 z_2 z_3 = 101$ we must consider three cases: $z = (10)^m$ with $m \geq 2$; $z = (10)^m 1$ with $m \geq 1$; and the case where $(10)^m 0$ is a prefix of z (with $m \geq 2$). In the first two cases $u = 0 \dots 0$. In the third case $0^{2m-2}1$ is a prefix of u and therefore $1u$ is not additive. □

Lemma 4. *Let* $1 \leq s \leq n$*. Let* $z \in \{0,1\}^{2n+1-s}$*. If* z *is left additive then* $f^n(1^s z) = 1$*. If* z *is right additive then* $f^n(z1^s) = 1$*.*

Proof. Direct from Lemma 3. Let us just consider the left additivity case. It is clear that the state of the leftmost cell (which is a 1 because $s \geq 1$) propagates to the right. Therefore, $f^n(1^s z) = 1$. □

Lemma 5. *If* $x'cy'$ *is additive, then*

1. *If* $|\alpha - \beta| \geq 1$ *then* $f^n(1^{n-\alpha}x', c, y'1^{n-\beta}) = 1$.
2. *If* $\alpha = \beta = k$ *then* $f^n(1^{n-\alpha}x', c, y'1^{n-\beta}) = f^k(x', c, y') = f^k(x', c, 0^k) + f^k(0^k, 0, y')$.

Proof.
1.- Let us assume, w.l.g., that $\alpha < \beta$. Consider $z = f^{\alpha+1}(1^{n-\alpha}x', c, y'1^{n-\beta})$. The number of bits of z is $2n + 1 - 2(\alpha + 1) = 2(n - \alpha - 1) + 1$. If we denote $z = z_{-(n-\alpha-1)} \dots z_0 \dots z_{(n-\alpha-1)}$ it follows that

$$z_{-(n-\alpha-1)} \dots z_0 = f^{\alpha+1}(1^{n-\alpha}x'_\alpha \dots x'_1 c y'_1 \dots y'_{\alpha+1}).$$

By Lemma 4 we conclude that $z_{-(n-\alpha-1)} \dots z_0 = 1^{n-\alpha}$. If $\alpha = n - 1$ then $z_0 = 1 = f^n(1^{n-\alpha}x', c, y'1^{n-\beta})$ and the result is concluded. If $\alpha < n - 1$ then $z_{-1}z_0 = 11$. Since such a wall of size two never changes we conclude that the central cell will always remain in state 1.

2.- If $k = n$ then it is direct. Suppose $k < n$. Consider the configuration $z = f^k(1^{n-k}x', c, y'1^{n-k})$. We conclude from Lemma 4 that $z = 1^{n-k}b1^{n-k}$, where $b = f^k(x', c, y') \in \{0, 1\}$. The results follows from the fact that $f(1, b, 1) = b$. □

Remark 1. The purpose of the following lemmas is to treat the case where $x'0y'$ is not additive (because $x'1y'$ is always additive). Therefore, we are interested in the case where an even length block of 0s between two consecutive 1s appear in $x'0y'$. In other words by recalling Notation 2, when $|l+r-1|$ is even or, equivalently, when $r \neq l \pmod 2$.

Lemma 6. *If $r \neq 0$, $l \neq 0$, $|l+r-1|$ is even and $l \geq r-1$, then*

$$f^n(x,0,y) = \begin{cases} f^n(1^{n-l+1}0^{l-1},0,y) & \text{if } l \geq r+3, \\ 1 & \text{if } |l-r| = 1. \end{cases}$$

Proof. Notice that $x_l \dots x_1 0 y_1 \dots y_r = 10^{l+r-1}1$. So $f^{\frac{l+r-1}{2}}(x_l \dots x_1 0 y_1 \dots y_r) = 11$. If $l > r$ then this 11 wall (through which information can not flow) will be located on the left side of the center cell. It follows that the final result will not depend on $x_n \dots x_{l+1}$ (if $l = n$ this is just the empty word). Then we can assume, w.l.g, that $x_n \dots x_{l+1} = 1^{n-l}$.

For the particular cases $l = r+1$ and $l = r-1$, the 11 wall will appear precisely in the center (and the result corresponds to 1). □

Lemma 7. *If $r \neq 0$, $l \neq 0$, $|l+r-1|$ is even and $l \leq r-3$, then*

$$f^n(x,0,y) = \begin{cases} f^\alpha(x',0,0^\alpha) & \text{if } r = \alpha+1, \\ 1 & \text{if } r \neq \alpha+1. \end{cases}$$

Proof. Since $r > l$, by the same argument used in the proof of Lemma 6, we know that the result does not depend on $y_{r+1} \dots y_n$ and we can therefore assume that $y_{r+1} \dots y_n = 1^{n-r}$. On the other hand, from Lemma 2, we can assume that $x_n \dots x_{\alpha+1} = 1^{n-\alpha}$. It follows from the two previous remarks that

$$f^n(x,0,y) = f^n(1^{n-\alpha}x_\alpha \dots x_1, 0, 0^{r-1}1^{n-r+1}).$$

Case $r = \alpha+1$. Let us denote

$$z_{-(n-\alpha)} \dots z_0 \dots z_{n-\alpha} = f^\alpha(x,0,y) = f^\alpha(1^{n-\alpha}x_\alpha \dots x_1, 0, 0^\alpha 1^{n-\alpha}).$$

Let us compute $z_{-2}z_{-1}z_0z_1z_2$ (if $\alpha = n-1$ we only consider $z_{-1}z_0z_1$). It follows that $z_{-2} = f^\alpha(11x_\alpha \dots x_3, x_2, x_1 0^{\alpha-1})$, $z_{-1} = f^\alpha(1x_\alpha \dots x_2, x_1, 0^\alpha)$, $z_1 = f^\alpha(x_{\alpha-1} \dots x_1 0, 0, 0^{\alpha-1}1)$, $z_2 = f^\alpha(x_{\alpha-2} \dots x_1 00, 0, 0^{\alpha-2}11)$. From Lemma 4, $z_{-2} = z_{-1} = z_1 = z_2 = 1$ (for $z_1 = 1$ and $z_2 = 1$ recall that $|l-r+1|$ is even). Therefore, the pattern $11z_011$ appears in the center. Since $z_0 = f^\alpha(x_\alpha \dots x_1, 0, 0^\alpha)$ the results follows (for the particular case $\alpha = n-1$ we have that $f^n(x,0,y) = f(z_{-1},z_0,z_1) = f(1,z_0,1) = z_0$ and the same conclusion is obtained).

Case $r > \alpha+1$. Let us denote

$z_{-(n-\alpha-1)} \dots z_0 \dots z_{n-\alpha} = f^{\alpha+1}(x,0,y) = f^{\alpha+1}(1^{n-\alpha}x_\alpha \dots x_1, 0, 0^{r-1}1^{n-r+1})$.

The result follows because $z_{-1}z_0 = 11$. In fact, $z_{-1} = f^{\alpha+1}(11x_\alpha \dots x_2, x_1, 0^\alpha)$ and $z_0 = f^{\alpha+1}(1x_\alpha \dots x_1, 0, 0^{\alpha+1})$. In both cases we apply Lemma 4.

Case $r < \alpha + 1$. Let us denote

$$z_{-(n-r)} \dots z_0 \dots z_{n-r} = f^r(x, 0, y) = f^r(1^{n-\alpha} x_\alpha \dots x_1, 0, 0^{r-1} 1^{n-r+1}).$$

The result follows because $z_0 z_1 = 11$. In fact, $z_0 = f^r(x_r \dots x_1, 0, 0^{r-1}1)$ and $z_1 = f^r(x_{r-1} \dots x_1 0, 0, 0^{r-2}11)$. In both cases we apply Lemma 4 (right additivity because $l + r - 1$ is even). In the particular case where $r = \alpha = n$ we have $f^n(x, 0, y) = z_0 = 1$. □

3.2 The protocols

3.2.1 When $c = 0$**.** We are going to define a one-round protocol $\mathcal{P}_0$ for the case where the central cell begins in state 0. Recall the Alice knows x and Bob knows y. $\mathcal{P}_0$ goes as follows. Alice sends to Bob α, l, and $a = f^\alpha(x', 0, 0^\alpha)$. The number of bits is therefore $2\lceil \log(n) \rceil + 1$.

If $l = 0$ then Bob knows (by definition of l) that $x = 0^n$ and he outputs $f^n(0^n, 0, y)$. If $r = 0$ his output depends on α. If $\alpha = n$ he outputs a (Lemma 1) and if $\alpha < n$ he outputs 1 (Lemma 5). We can assume now that neither l nor r are 0. The way Bob proceed depends mainly on the parity of $|l + r - 1|$.

Case $|l+r-1|$ **is odd.** In this case $x'0y'$ is additive and Bob can apply Lemma 5. In fact, if $|\alpha - \beta| \geq 1$ he outputs 1. If $\alpha = \beta = k$ he outputs $a + f^k(0^k, 0, y')$.

Case $|l + r - 1|$ **is even.** Bob compares r with l. If $l \geq r - 1$ then he applies Lemma 6. More precisely, he outputs $f^n(1^{n-l+1}0^{l-1}, 0, y)$ if $l \geq r + 3$ and 1 otherwise. If $l \leq r - 3$ then he applies Lemma 7. More precisely, he outputs $a = f^\alpha(x', 0, 0^\alpha)$ if $r = \alpha + 1$ and 1 otherwise.

Proposition 1. *$\mathcal{P}_0$ is a one-round f-protocol for $c = 0$ with cost $2\lceil \log(n) \rceil + 1$.*

3.2.2 When $c = 1$**.** We are going to define a one-round protocol $\mathcal{P}_1$ for the case where the central cell begins in state 1. Alice sends to Bob α and $a = f^\alpha(x', 1, 0^\alpha)$. The number of bits is therefore $\lceil \log(n) \rceil + 1$. Notice that $x'1y' \in \{0,1\}^{\alpha+\beta+1}$ is additive and therefore Bob applies Lemma 5. More precisely, if $\alpha \neq \beta$ then $f^n(x, 1, y) = 1$. On the other hand, if $\alpha = \beta = k$ then $f^n(x, 1, y) = f^k(x', 1, y')$ and Bob outputs $a + f^k(0, 1, y')$.

Proposition 2. *$\mathcal{P}_1$ is a one-round f-protocol for $c = 1$ with cost $\lceil \log(n) \rceil + 1$.*

4 Optimality

In this section we exhibit lower bounds for $d(M_f^{c,n})$, the number of different rows of $M_f^{c,n}$. If these bounds appear to be tight then, from Fact 1, they can be used for proving the optimality of our protocols.

4.1 Case $c = 0$.

Consider the following subsets of $\{0,1\}^n$. First, $S_3 = \{1^{n-3}000\}$. Also, $S_5 = \{1^{n-5}00000, 1^{n-5}01000\}$. In general, for every $k \geq 2$ such that $2k+1 \leq n$, we define $S_{2k+1} = \{1^{n-2k-1}0^{2k+1}\} \cup \{1^{n-2k-1}0^a10^b |\ a \text{ odd}, b \text{ odd}, b \geq 3, a + b = 2k\}$.

Lemma 8. *Let $x_n \dots x_1 \in S_{2k+1}$ and $\tilde{x}_n \dots \tilde{x}_1 \in S_{2\tilde{k}+1}$ with $k \neq \tilde{k}$. It follows that the rows of $M_f^{c,n}$ indexed by $x_n \dots x_1$ and $\tilde{x}_n \dots \tilde{x}_1$ are different.*

Proof. We can first easily prove (by induction on n) that every $z_n \dots z_1 \in \{0,1\}^n$ satisfies $f^n(z_n \dots z_1, 0, z_1 \dots z_n) = 0$. Let $x_n \dots x_1 \in S_{2k+1}$ and $\tilde{x}_n \dots \tilde{x}_1 \in S_{2\tilde{k}+1}$ (with $k \neq \tilde{k}$). From Lemma 5, $f^n(x_n \dots x_1, 0, \tilde{x}_1 \dots \tilde{x}_n) = f^n(\tilde{x}_n \dots \tilde{x}_1, 0, x_1 \dots x_n) = 1$. □

Lemma 9. *Let $x = x_n \dots x_1$, $\tilde{x} = \tilde{x}_n \dots \tilde{x}_1 \in S_{2k+1}$ with $x \neq \tilde{x}$. It follows that there exists $y = y_1 \dots y_n \in \{0,1\}^n$ such that $f^n(x,0,y) \neq f^n(\tilde{x},0,y)$.*

Proof. Assume, w.l.g., that for some odd number $b \geq 3$ the word 10^b is a suffix of x while 0^{b+2} is a suffix of $\tilde{x}$ (i.e., such b can be at most $n-4$). Let $y = 0^{b-3}101^{n-b+1}$. Let $z_{-(n-b+1)} \dots z_0 \dots z_{n-b+1} = f^{b-1}(x,0,y)$. Then $z_2 = f^{b-1}(0^{2b-5}1011)$ and $z_{-2} = f^{b-1}(10^{b-2},0,0^{b-1})$. It is direct that $z_{-2} = 1$. On the other hand, from Lemma 4 (right additivity), we know that $z_2 = 1$. For $z_{-1}z_0z_1$ notice that $z_{-1}z_0z_1 = f^{b-1}(0^{2b-2}101)$. In this case the dynamics is such that the configuration 0^*101 reappears every 2 steps. Since $b-1$ is even we conclude that $z_{-1}z_0z_1 = 101$. So we have proven that the pattern 11011 appears in the center and therefore $f^n(x,0,y) = 0$.

Let $\tilde{z}_{-(n-b-2)} \dots \tilde{z}_0 \dots \tilde{z}_{n-b-2} = f^{b+2}(\tilde{x},0,y)$. Then $\tilde{z}_1 = f^{b+2}(0^{2b-1}101111)$ and $\tilde{z}_0 = f^{b+2}(0^{b+2},0,0^{b-3}10111)$. From Lemma 4, $\tilde{z}_0 = \tilde{z}_1 = 1$ and therefore the wall 11 appears in the center. Since this means that $f^n(\tilde{x},0,y) = 1$, the lemma is proven. □

Proposition 3. *The cost of any one-round f-protocol for $c = 0$ is at least $2\lceil \log(n) \rceil - 5$.*

Proof. From Lemmas 8 and 9, the number of different rows in $M_f^{0,n}$ is $\sum_{3 \leq 2k+1 \leq n} |S_{2k+1}| = \sum_{i=1}^{\lceil \frac{n}{2} \rceil - 1} i$. For sufficiently large n the sum is lower bounded by $\frac{1}{16}n^2$. Therefore $d(M_f^{0,n}) \geq \lceil 2\log(n) - 4 \rceil \geq 2\lceil \log(n) \rceil - 5$. □

4.2 Case $c = 1$.

Proposition 4. *The cost of any one-round f-protocol for $c = 1$ is at least $\lceil \log(n) \rceil$.*

Proof. Consider the set $T = \{1^{n-k}0^k | 1 \leq k \leq n\}$. All we need to prove is that the rows indexed by any two different strings in T are different (because $|T| = n$).

Let $x = 1^{n-a}0^a$ and $\tilde{x} = 1^{n-\tilde{a}}0^{\tilde{a}}$ with $1 \leq a < \tilde{a} \leq n$. It is easy to prove (by induction on n) that $f^n(x,1,0^a1^{n-a}) = f^n(\tilde{x},1,0^{\tilde{a}}1^{n-\tilde{a}}) = 0$. It remains to prove that $f^n(x,1,0^{\tilde{a}}1^{n-\tilde{a}}) = f^n(\tilde{x},1,0^a1^{n-a}) = 1$.

By Lemma 5 we directly conclude that $f^n(x,1,0^{\tilde{a}}1^{n-\tilde{a}}) = 1$ except for the case when $\tilde{a} = a+1$ and a is odd. Let us therefore treat this last case now. First notice that $f^{a+1}(1^{n-a}0^a,1,0^{a+1}1^{n-a-1}) = 1^{n-a-1}f^{a+1}(10^a,1,0^{a+1})1^{n-a-1}$.

Therefore, by additivity (a is odd) and by the fact that $f(1,b,1) = b$ for all $b \in \{0,1\}$, the final result is

$$f^{a+1}(10^a,1,0^{a+1}) = f^{a+1}(00^a,1,0^{a+1}) + f^{a+1}(10^a,0,0^{a+1}) = 0 + 1 = 1. \quad \square$$

References

1. V. Bernardi, B. Durand, E. Formenti and J. Kari. *A new dimension sensitive property for cellular automata.* In *MFCS'04*, LNCS 3153, 416-426 (2004).
2. F. Blanchard and A. Maass. *Dynamical properties of expansive one-sided cellular automata.* Israel Journal of Mathematics, 99:149-174 (1997).
3. M. D'amico, G. Manzini, L. Margara. *On computing the entropy of cellular automata.* Theoretical Computer Science 290/3, 1629-1646 (2003).
4. M. Cook. *Universality in elementary cellular automata.* Complex Systems 15(1), 1-40, 2004.
5. C. Durr, I. Rapaport and G. Theyssier. *Cellular automata and communication complexity.* Theoretical Computer Science 322/2, 355-368 (2004).
6. E. Formenti and P. Kurka. *A search algorithm for the maximal attractor of a cellular automaton.* In *STACS'07*, LNCS 4393, 356-366 (2007).
7. J.E. Hanson and J.P. Cruchfield. *Computational mechanics of cellular automata: An example.* Physica D, 103:169-189 (1997).
8. J. Kari. *The nilpotency problem of one-dimensional cellular automata.* SIAM Journal on Computing 21(3): 571-586 (1992).
9. P. Kurka. *Languages, equicontinuity and attractors in cellular automata.* Ergodic Theory and Dynamical Systems, 17:417-433 (1997).
10. E. Kushilevitz and N. Nisan. *Communication Complexity.* Cambridge University Press, 1997.
11. K. Lindgren and M.G. Nordahl. *Universal computation in simple one-dimensional cellular automata.* Complex Systems 4, 299-318 (1990).
12. C. Moore. *Predicting non-linear cellular automata quickly by decomposing them into linear ones.* Physica D 111, 27-41 (1998).
13. J. Nagler and J.Ch. Claussen. $1/f^{\alpha}$ *spectra in elementary cellular automata and fractal signals.* Physical Review E 71, 067103 (2005).
14. M. Nasu. *Nondegenerate q-biresolving textile systems and expansive automorphisms of onesided full shifts.* Transactions of the American Mathematical Society, 358:871-891 (2006).
15. T. Neary and D. Woods. *P-completeness of cellular automaton Rule 110.* In *ICALP'06*, LNCS 4051, 132-143 (2006).
16. J. von Neumann. *The theory of self reproducting cellular automata.* University of Illinois Press, Urbana, Illinois, 1967.
17. N. Ollinger. *The quest for small universal cellular automata.* In *ICALP'02*, LNCS 2380, 318-329 (2002).
18. D. Regnault. *Directed percolation arising in stochastic cellular automata.* To appear in *MFCS'08*, 2008.
19. B. Srisuchinwong, T.A. York and Ph. Tsalides. *A symmetric cipher using autonomous and non-autonomous cellular automata.* In *Proc. of Global Telecommunication Conference*, IEEE Vol. 2, 1172–1177 (1995).
20. H. Umeo, K. Morita and K. Sugata. *Deterministic one-way simulation of two-way real-time cellular automata and its related problems.* Information Processing Letters 14(4): 158-161 (1982).
21. S. Wolfram. *Universality and Complexity in Cellular Automata.* Physica D 10, 1–35 (1984).
22. S. Wolfram. *A New Kind of Science.* Wolfram Media, Illinois, 2002.
23. A.C. Yao. *Some complexity questions related to distributed computing.* In *Proc. of 11th ACM Symposium on Theory of Computing*, 209–213, 1979.