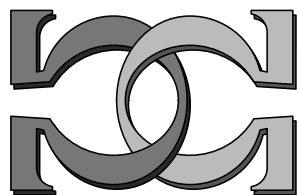
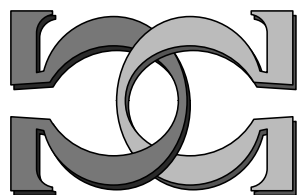
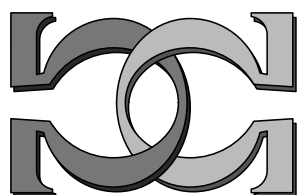


**CDMTCS  
Research  
Report  
Series**



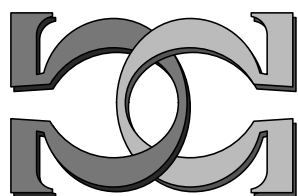
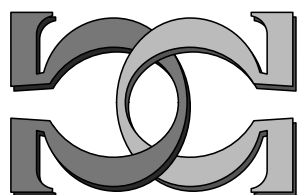
**The Underlying Optimal  
Protocol of Rule 218 Cellular  
Automaton**



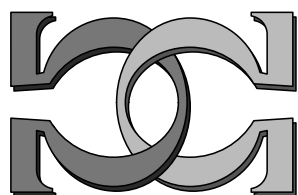
**Eric Goles<sup>1</sup>, Cedric Littleand<sup>1</sup>,  
Ivan Rapaport<sup>2</sup>**

<sup>1</sup>Universidad Adolfo Ibáñez, Chile

<sup>2</sup>Universidad de Chile, Chile



CDMTCS-314  
November 2007



Centre for Discrete Mathematics and  
Theoretical Computer Science

# The underlying optimal protocol of rule 218 cellular automaton \*

Eric Goles<sup>1</sup>, Cedric Little<sup>1</sup> and Ivan Rapaport<sup>2</sup>

<sup>1</sup>Facultad de Ciencias y Tecnologías, Universidad Adolfo Ibáñez, Chile

<sup>2</sup>Departamento de Ingeniería Matemática and Centro de Modelamiento Matemático, Universidad de Chile.

## Abstract

## 1 Introduction

Formally, an elementary cellular automaton (CA) is defined by a local function  $f : \{0, 1\}^3 \rightarrow \{0, 1\}$ , 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 [7, 8]). Sometimes, instead of expliciting function  $f$ , we refer to  $f_\omega$ .

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. In fact, for  $n = 1$  we define  $f^1(z_{-1}, z_0, z_1) = f(z_{-1}, z_0, z_1)$  and for  $n \geq 2$ :

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

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. In order to achieve this crucial goal we perform 2 steps.

**First step.** We represent  $f^n$  as two families of  $0-1$  matrices depending on whether the central cell begins in state  $c = 0$  or  $c = 1$ . These 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  $n = 1$ , defines completely 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.

---

\*Partially supported by Programs Conicyt “Anillo en Redes” (I.R.), Fondap on Applied Mathematics (I.R.) and Fondecyt 1070022 (E.G.).

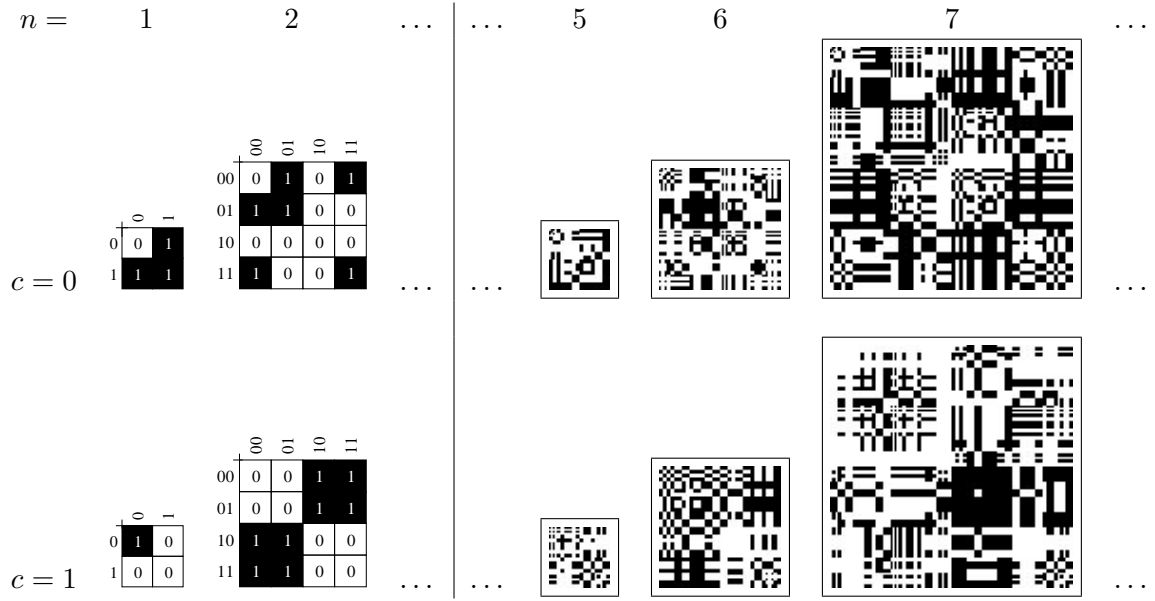


Figure 1: The two families of binary matrices  $M_{f_{54}}^{c,n}$  of Wolfram rule 54 cellular automaton.

**Second step.** In this step -which is obviously the most important- 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, let us consider rule 105. In Figure 2 we show on the left the space-time diagram of rule 105 for some arbitrary initial configuration, and on the right the matrix  $M_{f_{105}}^{0,6}$ . In contrast with the space-time diagram, the matrix looks simple. Indeed, we are going to mention the reason for such phenomenon later.

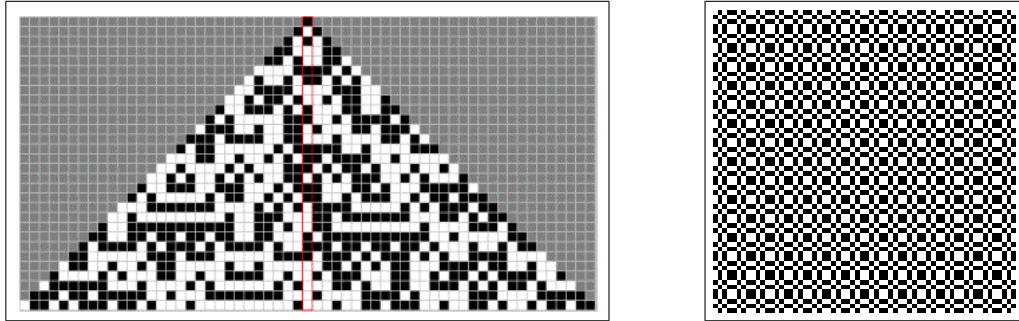


Figure 2: A space time diagram for rule 105 (left) and matrix  $M_{f_{105}}^{0,6}$  (right). In the diagram every row is a configuration and time goes upward (it shows only those cells on which the center cell depends).

Our working hypothesis is the following: the language of classical mathematics does not provide us with the flexibility we need in order to explain the structure of these matrices. We think that this is why the CAs (classification) problems encountered by the dynamical system community turn out to be so hard [ ].

We claim that the language of computer science is much more flexible and adequate for studying

the CAs. Therefore, instead of formulas or equations, we are going to exhibit simple protocols describing  $f^n$ . More precisely, 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  $f^n(x, 1, y)$ .

**Rule 218.** In a previous paper we began to explore the connections between CAs and communication complexity [2]. Nevertheless, in that work we were more interested in giving a formal classification than in understanding particular CAs behavior. In fact, if we denote by  $d(M)$  the number of different rows of a matrix  $M$ , then the only CAs we managed to explain were those we called *bounded* (where  $d(M_f^{c,n})$  was constant) and *linear* (where  $d(M_f^{c,n})$  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. Here we prove the existence of a CA for which  $d(M_f^{c,n})$  grows as  $\Theta(n^2)$ . This rule 218 CA is the following:

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

Notice that the global dynamics of rule 218 is represented by the two (beautiful) matrices of Figure 3. Notice also that rule 218 and rule 164 are the same (0s behave as 1s and viceversa).

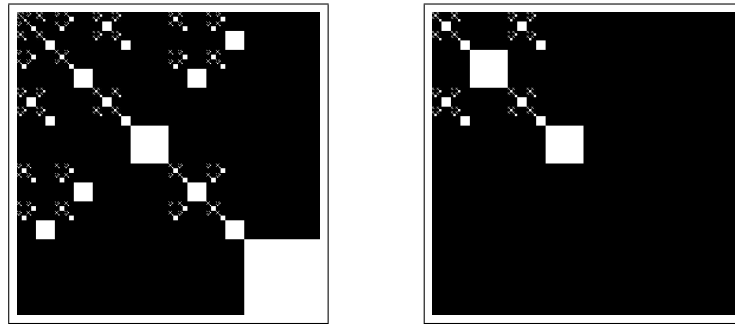


Figure 3:  $M_{f_{218}}^{0,9}$  (left) and  $M_{f_{218}}^{1,9}$  (right).

Linear and bounded rules are easy to explain. The goal of this work is to show that the underlying protocol of rule 218, on the other hand, is rather sophisticated. It will become clear in next section why this fact is related to the  $\Theta(n^2)$  behavior. Notice that this is the first rule for which we can prove such a behavior.

Rule 218 is very interesting. It is a kind of palindrome-recognizer and, despite the fact that it belongs to class 2 (according to Wolfram's classification [7, 8]), it mimics rule 90 (class 3) for very particular initial configurations. Authors in [4] were surprised when they found, "unexpectedly", that the rule exhibited  $1/f^\alpha$  spectra. Rule 218 has also been proposed as a symmetric cipher [6].

## 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 [9] 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** ([3]). *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:  $f(a, b, c) = a + b + c$  (the sum is mod 2). This is an additive rule and it satisfies the superposition principle. More precisely, for every  $x_n \dots x_1 \in \{0, 1\}^n$ ,  $\tilde{x}_n \dots \tilde{x}_1 \in \{0, 1\}^n$ ,  $y_1 \dots y_n \in \{0, 1\}^n$ ,  $\tilde{y}_1 \dots \tilde{y}_n \in \{0, 1\}^n$ ,  $c, \tilde{c} \in \{0, 1\}$ :

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

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 are going to denote  $f_{218}$  simply by  $f$ .

Notice that for 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.* Being additive is an invariant property: an additive configuration stays additive forever. Moreover, in this case, rule 218 behaves like the additive rule 90 (see Example 1). Therefore the superposition principle applies.  $\square$

**Definition 3.** Let  $\alpha$  be the maximum index  $i$  for which  $x_i \dots x_1 c$  is additive. Let  $\beta$  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  $y' = y_1 \dots y_\beta \in \{0,1\}^\beta$ .

**Lemma 2.** If  $x'cy'$  is additive, then

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

*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  $x_{\alpha+1} \dots x_{\alpha-2s} = 10^{2s}1$ . It follows that  $f^s(10^{2s}1) = 11$ . Notice that a word 11 acts as a wall through which information does not flow. 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  $x_n \dots x_{\alpha+1} = 1^{n-\alpha}$ .  $\square$

**Definition 4.** A string  $z$  is called strongly additive if  $z = 0 \dots 0$  or if it is additive while  $1z$  is not.

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

*Proof.* First we need to prove that the block of 1s moves to the right (see Figure 4 (left)). 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 some 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 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  $z_1 = 1$ . If  $z_2 = 1$  then  $u_1 = 1$  and  $1u$  is not additive. So we can assume that  $z_2 = 0$ . If  $z_3 = 0$  then again  $u_1 = 1$  and therefore can assume that  $z_3 = 1$  (see Figure 4 (right)). This means that it remains to consider the subcase  $z = (10)^m 0^{2l} 1$  is a prefix of  $z$  (with  $m \geq 2$  and  $l \geq 1$ ). It follows that  $0^{2m-2}1$  is a prefix of  $u$  and therefore  $1u$  is not additive.

$\square$

**Lemma 4.** Let  $1 \leq s \leq n$ . If  $x = 1^s x'$  and if  $x'cy$  is additive (in particular  $y' = y$ ), then  $f^n(x, c, y) = 1$ .

*Proof.* Direct from Lemma 3. In fact,  $x_n = 1$  and it propagates to the right. Therefore,  $f^n(x, c, y) = x_n = 1$ .  $\square$

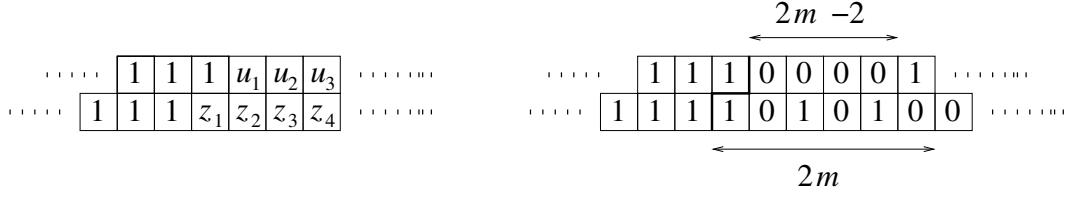


Figure 4:  $f(1^s z) = 1^s u$  (left) and  $0^{2m-2}1$  is a prefix of  $u$  (right).

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

**Definition 5.** *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  $r = 0$ .*

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

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

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

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

### 3.1 Case $c = 0$

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  $\alpha$ ,  $l$ , and  $a = f^\alpha(x', 0, 0^\alpha)$ . The number of bits is therefore  $2\lceil \log(n) \rceil + 1$ . Notice that if  $\alpha = n$  then Bob can easily decide: if  $y$  is also additive he outputs  $a + f^n(0^n, 0, y)$ ; otherwise he outputs 1 (Lemma ??). On the other hand, if  $\alpha < n$  and  $\beta = n$  he outputs 1. Therefore, we can assume from now on that neither  $x$  nor  $y$  are additive. The way Bob proceed depends mainly on the parity of  $|r - l + 1|$ .

**The case  $|r - l + 1|$  is odd.** In this case  $x'0y'$  is additive and Bob can apply Lemma 4. In fact, if  $|\alpha - \beta| \geq 1$  he outputs 1. If  $\alpha = \beta = k$  he outputs  $a + f^k(0^k, 0, y')$ .

**The case  $|r - l + 1|$  is even.** Bob computes  $r$  with  $l$ . If  $r \leq l + 1$  then he applies Lemma 6. More precisely, he outputs  $f^n(1^{n-l+1}0^{l-1}, 0, y)$  if  $r < l$  and 1 otherwise. If  $r \geq l + 3$  then he applies Lemma 7. More precisely, he outputs  $f^\alpha(x', 0, 0^\alpha)$  if  $r = \alpha + 1$  and 1 otherwise. We conclude the following proposition.

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

### 3.2 Case $c = 1$

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  $\alpha$  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 4. 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  $a + f^k(0, 1, y')$ . We conclude the following proposition.

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

## 4 Optimality

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

### 4.1 Case $c = 0$

Let us consider the following subsets of  $\{0, 1\}^n$ . First,  $S_1 = \{1^{n-1}0\}$ . Also,  $S_3 = \{1^{n-3}000, 1^{n-3}010\}$ . In general, for every  $k \geq 1$  such that  $2k + 1 \leq n$ , we define

$$S_{2k+1} = \{0^{2k+1}1^{n-2k-1}\} \cup \{1^{n-2k-1}0^a10^b \mid a \text{ odd}, b \text{ odd}, a + b = 2k\}.$$

**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 4,  $f^n(x_n \dots x_1, 0, \tilde{x}_1 \dots \tilde{x}_n) = f^n(\tilde{x}_n \dots \tilde{x}_1, 0, x_1 \dots x_n) = 1$ .  $\square$

**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.*  $\square$

**Proposition 3.** The cost of any one-round  $f$ -protocol for  $c = 0$  is at least  $2\lceil \log(n) \rceil - 4$ .

*Proof.* From Lemmas 8 and 9 we know that the number of different rows in  $M_f^{0,n}$  is at least

$$\sum_{1 \leq 2k+1 \leq n} |S_{2k+1}| = \sum_{i=1}^{\lceil \frac{n}{2} \rceil} i \geq \frac{1}{8}n^2.$$

Therefore  $d(M_f^{0,n}) \geq \lceil \log(2n - 3) \rceil \geq 2\lceil \log(n) \rceil - 4$ .  $\square$

### 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  $T = \{1^{n-k}0^k \mid a + b = n, 1 \leq k \leq n\}$ . All we need to prove now is that the rows indexed by any two different strings in  $T$  are different (the result would follow because  $|T| = n$ ). Let  $x = 1^{n-a}0^a$  and  $x' = 1^{n-a'}0^{a'}$  with  $1 \leq a, a' \leq n$  and  $a \neq a'$ . Notice first that it is easy to prove (by induction on  $n$ ) that  $f^n(1^{n-a}0^a, 1, 0^a1^{n-a}) = f^n(1^{n-a'}0^{a'}, 1, 0^{a'}1^{n-a'}) = 0$ . On the other hand, by Lemma 4,  $f^n(1^{n-a}0^a, 1, 0^a1^{n-a'}) = f^n(1^{n-a'}0^{a'}, 1, 0^a1^{n-a}) = 1$ .  $\square$



## References

- [1] M. Delorme and J. Mazoyer. *Cellular Automata*. Kluwer Academic Publishers, 1999.
- [2] C. Durr, I. Rapaport and G. Theyssier. *Cellular automata and communication complexity*. Theoretical Computer Science 322/2 (2004), 355-368.
- [3] E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge University Press, 1997.
- [4] J. Nagler and J.Ch. Claussen.  $1/f^\alpha$  spectra in elementary cellular automata and fractal signals. Phys. Rev. E 71, 067103 (2005).
- [5] J. von Neumann. *The theory of self reproducing cellular automata*. University of Illinois Press, Urbana, Illinois, 1967.
- [6] 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).
- [7] S. Wolfram. *Universality and Complexity in Cellular Automata*. Physica D 10, 1–35 (1984).
- [8] S. Wolfram. *A New Kind of Science*. Wolfram Media, Illinois, 2002.
- [9] A.C. Yao. Some complexity questions related to distributed computing. In *Proc. of 11th ACM Symposium on Theory of Computing*, 209–213, 1979.