Traced communication complexity of cellular automata

Eric Goles

Facultad de Ingeniería y Ciencias
Universidad Adolfo Ibañez
Santiago, Chile

Pierre Guillon

DIM, CMM (UMI 2807 CNRS)
Universidad de Chile
Santiago, Chile

Ivan Rapaport

Abstract

We study cellular automata with respect to a new communication complexity problem: each of two players know half of some finite word, and must be able to tell whether the central cell will ever reach a particular state. We present some links with classical dynamical concepts, and give the asymptotic communication complexity of many elementary cellular automata.

Keywords: Cellular automata, communication complexity

Cellular automata (CAs) were introduced in the fifties in order to represent natural complex systems. It was soon studied as a computational model, especially for parallel computation. Indeed, it can be seen as a wide network of small machines communicating locally.

Introduced in [7], the communication complexity of a function $f$ measures how many bits must be exchanged between two machines in order to compute $f$. This approach was adapted to the context of CAs in [1] and gave interesting results in [2,3,4] that help better understand some CA rules. It gave a notion of complexity of the rule which is different from the classical ones on dynamical systems. In particular, we can see that bipermutive CAs, which show strongly chaotic behaviors (for many reasonable definitions), are simple with respect to this complexity measure.

1 Email: eric.chacc@uai.cl
2 Email: pguillon@dim.uchile.cl
3 Email: rapaport@dim.uchile.cl
4 Partially supported by the ECOS-Sud Project, Fondecyt 1070022 and 1090156 and BASAL project CMM-U, de Chile.

This paper is electronically published in
Electronic Notes in Theoretical Computer Science
URL: www.elsevier.nl/locate/entcs
In this article, we address a new variant of the communication complexity problems, which involves more links with dynamical chaos.

1 Definitions

For a finite word $u \in \{0,1\}^*$, we note $|u|$ its length. A configuration is a biinfinite word $x \in \{0,1\}^\mathbb{Z}$. We denote $x_{[i,k]}$ (resp. $x_{[i,k]}$) the finite word $x_i \ldots x_k$ (resp. $x_i \ldots x_{k-1}$).

Subshifts.

A twosided subshift is a subset $\Sigma \subseteq \{0,1\}^\mathbb{Z}$ of configurations that avoids some set $\mathcal{F} \subseteq \{0,1\}^*$ of patterns, i.e. $\Sigma = \{ x \in \{0,1\}^\mathbb{Z} \mid \forall i,k, x_{[i,k]} \notin \mathcal{F} \}$. Similarly, we define onesided subshifts. We expect the reader to distinguish between onesided and twosided subshifts from the context. The language of some subshift $\Sigma$ is the set $\mathcal{L}(\Sigma) = \{ u \in \{0,1\}^* \mid \exists x \in \Sigma, x_{[0,|u|]} = u \}$ of its authorized patterns. We denote $\mathcal{L}_n(\Sigma) = \mathcal{L}(\Sigma) \cap \{0,1\}^n$.

The entropy of some subshift $\Sigma$ is the limit $\lim_{n \to \infty} \frac{\log |\mathcal{L}_n(\Sigma)|}{n}$ (all logarithms are assumed binary). As an example, the full shift $\{0,1\}^\mathbb{Z}$ has maximal entropy 1. There are many subshifts of null entropy, such as the empty subshift or the subshift of only forbidden pattern 10.

Communication complexity.

Let $f$ a binary map defined over the product $X \times Y$. Take $(x,y) \in X \times Y$, and consider that two persons, Alice and Bob, are given $x$ and $y$ respectively, and must compute the value $f(x,y)$ by communicating as little as possible.

The multi-round communication complexity (CC) of a function $f$ is the cost of the best exchange protocol between the two players. A protocol is leftsided if only Alice sends information to Bob. The left (one-round) communication complexity is the worst case number of bits Alice needs to send to allow Bob to compute. We can define a rightsided protocol and the right communication complexity similarly. Of course, the multi-round CC is smaller than the two one-round CC.

A function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ can be represented by the square matrix $M_f$ of size $2^n$ such that $M_f^i \sum_{u \in \Sigma, \sum_{v \in \Sigma} u 2^i = f(u_0, \ldots, u_{n-1}, v_0, \ldots, v_{n-1})$. Then it is known (see for instance [5]) that the left (resp., right) CC of $f$ can be seen as the logarithm of the number of distinct lines (resp., columns) in the matrix $M_f$. Moreover, the multi-round CC is conjectured in [6] to be polylogarithmic in the rank of the matrix. We will see some examples of such matrices, which help get a visual intuition of the asymptotic CC.

A set $S \subseteq X \times Y$ is a fooling set if for every two distinct pairs $(x_1, y_1), (x_2, y_2) \in S$, $f(x_2, y_1) \neq f(x_1, y_1) = f(x_2, y_2) \neq f(x_1, y_2)$.

Remark 1.1 [5] If a function $f$ admits a fooling set $S$, then its multi-round CC is lower-bounded by $\log |S|$.

Let $\Sigma$ a subshift and $f$ a binary map defined over $\bigcup_{n \in \mathbb{N}} \mathcal{L}_{2n+1}(\Sigma)$. The multi-round (resp, left, right, one-round) communication complexity (CC) of $f$ is the...
function mapping each \( n \in \mathbb{N} \) to the maximum CC between the two functions parametrized by \( a \in \{0, 1\} \):

\[
\tilde{f}_a : \left\{ (u, v) \in \mathcal{L}_n(\Sigma)^2 \mid uav \in \mathcal{L}(\Sigma) \right\} \rightarrow \{0, 1\} \\
(u, v) \mapsto f^n(uav)
\]

Clearly, if \( \Gamma \) is a subshift included in \( \Sigma \), then the CC of \( f|_{\mathcal{L}(\Gamma)} \) is upper bounded by the CC of \( f \).

**Remark 1.2** The multi-round CC of a function over the language of some subshift \( \Sigma \) is bounded by the function \( n \mapsto \log |\mathcal{L}_n(\Sigma)| \). In particular, if \( \Sigma \) has entropy \( \alpha > 0 \), then the CC is asymptotically at most \( \alpha n \).

**Traced communication complexity.**

CAs consist in sequences of cells with states evolving according to their neighbors. We restrict our study to elementary CAs: dimension 1, binary states and nearest neighbors. In this context, a *partial cellular automaton* (PCA) on some subshift \( \Sigma \) is a function \( f : \mathcal{L}_3(\Sigma) \rightarrow \{0, 1\} \) such that \( \forall x \in \Sigma \), the synchronous application \( (f(x_{i-1}x_i x_{i+1}))_{i \in \mathbb{Z}} \) is in \( \Sigma \). A *cellular automaton* (CA) \( f \) is a PCA on \( \Sigma = \{0, 1\}^\mathbb{Z} \). There are exactly 256 such CA, which can be referred to by the following canonical number: \( \sum_{a, b, c \in \{0, 1\}} f(abc)^{2^a + 2^b + c} \).

The local rule \( f \) of a PCA (or CA) on some subshift \( \Sigma \) can be canonically extended to any finite word \( u \in \mathcal{L}(\Sigma) \) by \( f(u)_i = f(u_{i-1}u_i u_{i+2}) \) for \( 1 \leq i < |u| - 1 \).

In [1,2,3], CC over CA has been studied as the CC of the function \( \tilde{f} \) that maps to each \( u \in \mathcal{L}_2n+1(\Sigma) \) the bit \( f^n(u) \). In other words, Alice and Bob want to know whether the central cell will get the value 1 after \( n \) steps. We will refer to this notion as classical CC.

In [4], some new problems, the so-called invasion and cycle length problems, were associated to the CAs. In the following, we are interested in the following problem: we require Alice or Bob to say whether the central cell will have the value 1 after \( n \) steps (the state of the central cell is initially 0). Formally, the traced CC of the PCA associates each \( n \in \mathbb{N} \) to the CC of the following function:

\[
\hat{f} : \left\{ (u, v) \in \mathcal{L}_n(\Sigma)^2 \mid u0v \in \mathcal{L}(\Sigma) \right\} \rightarrow \{0, 1\} \\
((u_i)_{-n \leq i \leq -1}, (v_i)_{1 \leq i \leq n}) \mapsto \begin{cases} 0 & \text{if } \forall j \leq n, f^j(u0v)_0 = 0; \\ 1 & \text{otherwise.} \end{cases}
\]

The figures in the next sections represent the matrices of the function \( \tilde{f} \) (assimilating white with red) superposed with the corresponding matrices of the function \( \hat{f} \) (assimilating red with black); in other words: red cells correspond to the words which evolution has reached state 1 but came back to state 0.

### 2 Simple communications

**One-sided rules.**

Similarly to the case of the classical CC, the traced CC of any one-sided CA, that is those that depend only on either left cells or right cells, is clearly constant.
- one of the two sides being completely able to compute the function \( \hat{f} \). We can be a little more precise.

A CA \( f \) is 0-leftsided if \( \forall a \in \{0,1\}, f(a00) = f(a01) \). Similarly, we define 0-rightsided CA. In such a CA, one side can compute the central cell until it reaches some 1: no communication is needed.

**Proposition 2.1** Any 0-onesided CA \( f \) has constant one-round traced CC.

**Proof.**

- If it is 0-leftsided, Alice can compute when the first 1 will appear in the central cell and give the answer to Bob.
- If it is 0-rightsided, she does not say anything to Bob; he will be able to find the answer by himself.

The previous proposition can be applied to the 64 CA which number can be written as \( u_7u_6u_4u_3u_2u_0u_0 \) in base 2.

![Matrix of the rule 143](image)

**Spreading states.**

A state \( a \in \{0,1\} \) is quiescent for CA \( f \) if \( f(aaa) = a \). We say that 1 is left (resp., right) weakly spreading for CA \( f \) if \( f(001) = 1 \) (resp., \( f(100) = 1 \)). It is left (resp., right) semi-strongly spreading if, besides, \( f(101) = 1 \).

We can see that if 1 is left and right semi-strongly spreading and 0 is quiescent, then \( \hat{f}(u0v) = 1 \) if and only if \( u0v \) contains some 1. We can generalize a little.

**Proposition 2.2** If 1 is left semi-strongly spreading, then the right traced CC is constant.

**Proof.**

- First assume that 0 is quiescent. Then, if Bob has a word of the form \( v = 0^i1u_{i+2} \ldots u_n \), with \( i < n \), and Alice some word \( u \), then \( \hat{f}(u0v) = 1 \). Otherwise, he says (in one bit) to Alice he only has \( 0^n \) and she can compute \( \hat{f}(u0^n+1) \).
- Now, assume that \( f(000) = 1 \), \( n \geq 2 \), Bob has a word \( v_1 \ldots v_n \), and Alice some word \( u_{-n} \ldots u_{-1} \). If \( v_1 = 1 \), then \( f(u_{-1}0v_1) = 1 \). If \( v_1 = 0 \), then whatever \( v_2 \), \( f(0v_1v_2) = 1 \), which leads, in the case when \( f(u_{-1}0v_1) = 0 \), to \( f^2(u_{-2}u_{-1}0v_1v_2) = 1 \). As a result \( \hat{f} \) is constantly equal to 1 as soon as \( n \geq 2 \).□

Similarly, if 1 is right semi-strongly spreading, then the left traced CC is constant. These two cases apply to the 96 CA which number can be written either as
Stagnating bunches.

A word $u \in \{0, 1\}^*$ is stagnating for CA $f$ if $\forall a, b \in \{0, 1\}, f(aub) = u$.

Note that if 0 is stagnating, then it is both 0-leftsided and 0-rightsided; we have seen that the one-round CC is constant. We can generalize to the following case.

**Proposition 2.3** If 0 is quiescent and 1 is neither left nor right weakly spreading, then the one-round traced CC are constant.

**Proof.**

• If $f(101) = 0$, then 0 is stagnating; this is a subcase of proposition 2.1.
• If $f(101) = 1$, then 00 is stagnating, but single 0s disappear. Hence $\hat{f}(uv) = 0$ if and only if $u_{-1} = 0$ or $v_{1} = 0$. The result of this test can be transmitted easily. 


**Remark 2.4** Any CA $f$ with stagnating 1 has a traced CC equal to its classical CC. Indeed, $\hat{f}$ is equal to $f$. For instance, the CA 222 has logarithmic traced CC.

### 3 Trace

Alice and Bob need to know only the first 1 appearing in the central cell. In some sense, in their computation, they can assume that the central cell is always 0.

For $n \in \mathbb{N}$, let $f_{-} : \{0, 1\}^{n+1} \rightarrow \{0, 1\}^n$ and $T_{f_{-}} : \{0, 1\}^{n} \rightarrow \{0, 1\}^n$

$$v \mapsto f(0v)$$

Similarly, we can define $T_{f_{-}}$ in Alice’s side.

\[ 5 \]
Intuitively, $T_{f-}(v)$ represents the evolution of the cell on the right of the central cell, assuming that the latter is always equal to 0. If this word has a short algorithmic complexity, then the right traced CC will be low.

**Proposition 3.1** The right traced CC is bounded by $\log|T_{f-}(\{0,1\}^n)|$.

**Proof.** If Alice has word $u$, Bob word $v$ and $J$ is such that $\forall j < J, f^j(u0v)_0 = 0$, then by an immediate recurrence, we have $\forall j < J, f^{j+1}(u0v) = f(T_{f-}(u)_j0T_{f-}(v)_j)$. It is sufficient that Bob transmits this word for Alice to compute (thanks to the cells of her side) the first 1 appearing in the central cell. \(\square\)

Obviously, the left traced CC is bounded by $\log|T_{f-}(\{0,1\}^n)|$.

This can be applied in two simple cases.

**Proposition 3.2** If $\forall a \in \{0,1\}, f(0a0) = f(0a1)$, then the right traced CC is constant.

**Proof.** If Bob has the word $v = v_1 \ldots v_n$, then it is sufficient for him to send to Alice the initial state $v_1$ of the right cell. Then Alice can compute $T_{f-}(v)$ since it does not depend on the other cells. From the previous remark, she can compute whether the central cell will reach the state 1. \(\square\)

Similarly, if $\forall a \in \{0,1\}, f(1a0) = f(0a0)$, then the left traced CC is constant.

The following CA are concerned by one of these two conditions but not the previous ones: 19, 27, 31, 95, 147, 155, 159, 223.

Matrix of the rule 159

The hypothesis corresponds to some kind of partial onesidedness. Either the state loops, or it remains the same. In the latter case, we can be a little bit more general if we allow logarithmic communications.

**Proposition 3.3** If $\exists a \in \{0,1\}, f(0a0) = f(0a1) = a$, then the right traced CC is at most logarithmic.

**Proof.** If Bob has the word $v$, then it is sufficient for him to send to Alice the first generation when some $a$ appears in $T_{f-}(v)$ ($|v| + 1$ if $a$ never appears). Indeed, Alice will then know entirely this word, and be able to calculate $\hat{f}$ from the remark above. \(\square\)

Similarly, if $\exists a \in \{0,1\}, f(0a0) = f(1a0) = a$, then the left traced CC is at most logarithmic.

Note that the previous case includes that, already seen, of CA having stagnating 0, or stagnating 1, as well as the following other CA: 41, 45, 105, 109, 169, 173, 233,
Entropy.

We define the trace of the PCA \( f \) on some (implicit) subshift \( \Sigma \) as the set \( \tau_f \) of infinite words \( (f^n(x_{[-n,n]}))_{n \in \mathbb{N}} \in \{0,1\}^\mathbb{N} \) for \( x \in \Sigma \). In other words, it contains the sequences of states that represent some evolution of the central cell with respect to the applications of the rule. It is a onesided subshift. A PCA is equicontinuous if its trace is a finite set.

The previous remarks give us the following rough bound.

**Proposition 3.4** The one-round traced CC of the PCA \( f \) is bounded by the function \( n \mapsto \log |\mathcal{L}_n(\tau_f)| \).

**Proof.** Let \( \tau^n_{f-} = T_{f-}(\{0,1\}^n) \cap \mathcal{L}_n(\tau_f) \). Assume Bob has some word \( v \in \{0,1\}^n \). If \( T_{f-}(v) \notin \tau^n_{f-} \), then there is no \( u \in \{0,1\}^n \) such that \( \hat{f}(u0v) = 0 \). Bob can send a bit saying this. Otherwise, it can encode \( T_{f-}(v) \) as an element of \( \tau^n_{f-} \) (in at most \( \log |\mathcal{L}_n(\tau_f)| \) bits) and send it to Alice. As well as in Proposition 3.1, Alice can then compute the value of \( \hat{f} \). \( \square \)

In particular, we can see that, like for the classical CC, equicontinuous CA, which are known to have a bounded trace language, have a constant traced CC, and that if \( \tau_f \) has entropy \( \alpha > 0 \), then the one-round traced CC is asymptotically less than \( \alpha n \).

Moreover, a characteristic quantity of PCA is their entropy, which can be somehow seen as the limit of the entropy of the traces when we widen the range of observed cells instead of taking only the central state. Hence, the previous proposition allows to bound the traced CC by the entropy of the CA.

## 4 Permutative rules

A PCA \( f \) on subshift \( \Sigma \) is right permutative if for any \( a, b \in \{0,1\} \), such that \( ab0, ab1 \in \mathcal{L}_3(\Sigma) \), \( f(ab0) \neq f(ab1) \), \( i.e. \) it does a permutation on the last variable, whatever the fixed other cells. Similarly, we can define left permutativity. A CA is bipermutive if it is both left and right permutative.

Let \( f \) a CA on the subshift \( \Sigma \), and \( n \in \mathbb{N} \). If \( f \) is right permutive, then for any word \( u \in \mathcal{L}_n(\Sigma) \), the application \( R_{f,u} : \mathcal{L}_n(\Sigma) \to \mathcal{L}_n(\Sigma) \) defined by \( v \mapsto (f^j(u0v))_{j \leq n} \) is injective (where \( f^j(u0v)_0 \) represents the central cell of the word). We define the application \( L_{f,u} \) in a similar way; it is injective if \( f \) is left permutative.
This remark helps us build large fooling sets: this kind of CA is very complex with respect to traced CC.

**Proposition 4.1** Let $f$ be a bipermutive PCA on some subshift $\Sigma$ such that for any $n \in \mathbb{N}$, $L(\Sigma) \supset X_n0Y_n$, with $X_n,Y_n \subset L_n(\Sigma)$ and for any $u \in X_n$, $R_{f,u}(Y_n) \ni 0^{n+1}$. Then the multi-round traced CC of $f$ is greater than $n \mapsto \log |X_n|$.

**Proof.** By bipermutivity, for each $n \in \mathbb{N}$ and each $u \in X_n$, there is a unique word $R^{-1}_{f,u}(0^{n+1})$ of $Y_n$ which, juxtaposed to $u$, gives only 0s in the central cell. Let us show that $S = \{(u, R^{-1}_{f,u}(0^{n+1})) | u \in X_n\}$ is a fooling set for the corresponding function on $\{(u,v) \in L_n(\Sigma)^2 | u0v \in L(\Sigma)\}$. Indeed, for any $(u_1,v_1) \in S$, by construction, $R_{f,u_1}(v_1) = 0^{n+1}$, i.e. $f(u_10v_1) = 0$. Moreover, for any other couple $(u_2,v_2) \in S$, the right permutivity gives $R_{f,u_1}(v_2) \neq R_{f,u_1}(v_1)$, and the left permutivity gives $L_{f,v_1}(u_2) \neq L_{f,v_1}(u_1)$, i.e. a 1 appears in each of the two words $R_{f,u_1}(v_2)$ and $L_{f,v_1}(u_2)$. Since $|S| = |X_n|$ we can conclude thanks to remark 1.1.

We will now see two particular cases where the hypothesis of the previous proposition is satisfied. First, notice that if $\Sigma$ is the full shift, then the hypothesis over the language is trivially satisfied.

**Corollary 4.2** Any bipermutive CA on $\{0,1\}^\mathbb{Z}$ has linear multi-round traced CC.

**Proof.** To apply proposition 4.1, it is enough to take $X_n = Y_n = \{0,1\}^n$; in this context, if the applications $R_{f,u}$, for $u \in X_n$, are injective, then they are bijective onto $\{0,1\}^n$.

This implies that CA 90, 150, 105, 165 have linear (i.e. maximal) traced CC.

**Legal rules.**

Let $\overline{u}$ denote the mirror word $u_{|u|-1} \ldots u_0$ of $u \in \{0,1\}^*$. If $X$ is a set of finite words, $\overline{X}$ is the set of all the corresponding mirror words. A PCA $f$ on subshift $\Sigma$ is 0-legal if for any $u \in L_3(\Sigma)$, $\overline{u} \in L(\Sigma)$ and $f(\overline{u}) = f(u)$, and for any $a \in \{0,1\}$ such that $a0a \in L(\Sigma)$, $f(a0a) = 0$.

**Lemma 4.3** For any 0-legal PCA $f$, any $n \in \mathbb{N}$, any $u$ such that $u0\overline{u} \in L_{2n+1}(\Sigma)$, and any $j \leq n$, $f^j(u0\overline{u})_0 = 0$.

**Proof.** This comes from an immediate recurrence: if $n > 0$, then $f(u0\overline{u})$ has a central 0 and remains a symmetric word, from the symmetry of the rule.

The previous lemma allows in this context to establish an equivalence between the problem of traced CC and the classical equality test of two bit sequences, which is known to have maximal CC.
Proposition 4.4 Let $f$ be a bipermutive 0-legal PCA on some subshift $\Sigma$ such that for any $n \in \mathbb{N}$, $\mathcal{L}_{2n+1}(\Sigma)$ contains some sublanguage of the form $X_n0\overline{X}_n$. Then the multi-round traced CC of $f$ is greater than $n \mapsto \log |X_n|$. 

Proof. To apply Proposition 4.1, it is enough to take $Y_n = \overline{X}_n$. The hypothesis is satisfied thanks to lemma 4.3. 

Corollary 4.5 The CA 18, 26, 146, 154, 218 have linear multi-round traced CC. 

Proof. Note that these rules are equal to the bipermutive rule 90 except on patterns 011, 110, 111. Define $\Sigma$ as the set of configurations avoiding the pattern 11 and the patterns $10^k+1$, for $k \in \mathbb{N}$. It can be seen that $\Sigma$ is stable by the synchronous application of CA 90, hence by any of these CA. 

Moreover, for $k \in \mathbb{N}$, define $X_k = (0100 + 0001)^k$. Note that $X_k0\overline{X}_k \subset \mathcal{L}_{8k+1}(\Sigma)$. From proposition 4.4, the traced CC of order $4k$ of $f|_\Sigma$ is greater than $\log |X_k| = k$. 

Matrix of the rule 146 

Unfortunately, we do not know such practical subsystems for the three other bipermutive elementary CA that would allow other applications of proposition 4.4. 

5 Conclusion 

In this paper, we have addressed a new problem of communication complexity over cellular automata, to get some clues about the information flows. Indeed, it can help understand CA behaviors by exhibiting what communication is needed to achieve its computation. 

We have treated most elementary cellular automata; some of the remaining ones look experimentally easy, others rather mysterious, such as 22 or some CA where 1 is weakly but not semi-strongly spreading. 

Note that this problem can be restated as wondering whether or not the word corresponding to the evolution of the central cell is $0^{n+1}$. Our proofs on bipermutive CA and on the trace of a neighboring cell can be adapted to the case of any other specified word $u \in \{0,1\}^{n+1}$: for instance Alice and Bob want to know if the central cell has alternating 0s and 1s. 

This restatement is still valid when taking a larger alphabet. In that setting, bipermutivity might be replaced by positive expansivity, which corresponds exactly to our condition of injectivity of the functions $L_{f,u}$. 

Widening the radius would introduce the problem of considering a single cell, but thanks to synchronism, we can hope that the evolution of each cell cannot be much easier to compute than the evolution of a group of cells. 

To sum up, this new CC issue presents several links with respect to some classical notions of chaos (entropy, expansivity), but maybe less links with computation than
the classical one since it shall be easy to exhibit intrinsically universal CA which have a trivial traced CC.

References


