The Reachability problem for finite cellular automata

Andrea Clementi
Centre Universitaire d’Informatique
University of Genève
Genève (CH)

Russell Impagliazzo
Department of Computer Science
University California of San Diego
La Jolla - CA (USA)

( Part of the results presented in this paper have already appeared in [5] )

Key words: Computational complexity, Cellular automata, Reachability problems, Interactive proofs.

Abstract

We investigate the complexity of the Configuration REachability Problem (CREP) for two classes of finite weakly predictable cellular automata: the invertible and the additive ones. In both cases we prove that CREP belongs to the “Arthur-Merlin” class CoAM[2].

1 Introduction and preliminaries

There is a renewal of interest in the theory of cellular automata since they represent one of the few parallel computing models which explicitly consider the ultimate impact of fundamental physical laws [14].

Informally, a cellular automaton consists of an array of finite-state automata (also called cells) locally interconnected which evolve in a synchronous discrete way. Thus the local interactions uniquely determine a global function $F$ acting on the space $\Sigma$ of all possible configurations and the behaviour of the system can be described by the equation: $X^t = F(X^{t-1}) = F^t(X^0)$ ($X^t \in \Sigma$ for any $t \geq 0$). In this paper we study the Configuration REachability Problem (in short CREP) which consists of deciding, given a cellular automaton and a pair $\langle X, Y \rangle$ of configurations, whether an instant $t \geq 0$ exists such that $F^t(X) = Y$. CREP is, in general, $PSPACE$-complete (see for example [12]), thus, cellular automata are considered, in general, unpredictable. However, CREP’s computational complexity may vary greatly depending
on the local interactions considered in input (see [8, 12]). A cellular automaton is called weakly predictable if $F'(X)$ is polynomial-time computable (i.e. in time $O((\log t)|X|^k)$, where $k$ is a positive constant) and predictable if CREP is decidable in polynomial-time. The weakly predictable and predictable notions formalize the concept of *computationally reducible dynamical systems* first introduced in [15] and, successively, in [3]. For weakly predictable cellular automata, CREP is $NP$-complete [12].

We study CREP for the class INV of weakly predictable cellular automata having invertible global functions (for an introductory survey on invertible cellular automata see [14] and, in particular, for crypto applications of constructive subsets of INV see [16]). It has been recently shown [5] that CREP, even when restricted to invertible additive rules (namely, a particular subset of INV), is unlikely to be polynomial-time decidable (unless the vectorial version of the Discrete Log problem is polynomial-time computable). However, by using concepts from the theory of interactive proof systems (see [1, 7]), we give an interactive protocol for the complement of CREP proving that $\text{CREP} \in \text{CoAM}[2]$, where $\text{AM}[k]$ is the “Arthur-Merlin” complexity class defined in [1, 7]. This also implies that CREP, for the class INV, cannot be $NP$-complete unless the polynomial-time hierarchy collapses. Another important class of weakly predictable cellular automata is the additive one which was first introduced in [11] and successively studied in [9]. However, the complexity of CREP, for this class, was still unknown. As final result of this paper, we describe the main arguments for adapting the interactive protocol to additive cellular automata. Thus, for this class, we obtain the same complexity results as those for the class INV.

An overall interpretation for the results shown in this paper confirms that the approach proposed in [4, 3, 12, 15] and based on the analysis of CREP, can give interesting consequences in the theory of cellular automata. It turns out that the computational complexity of CREP can be considered as a “good” measure for classifying cellular automata. Under this point of view, cellular automata, belonging to the INV class or to the additive one, can be considered “more” predictable than general weakly predictable cellular automata. More precisely, as far as the global structure of the weakly predictable class is considered, these two classes play the same role of that of the $NP$-Intermediate ($NPI$) languages (like the *Graph Isomorphism Problem* [10]) with respect to the structure of $NP$.

### 1.1 Preliminaries

The formal definition of a cellular automaton (in short *ca*) can be given as a quintuple $A = (d, n, Q, N, f)$ where:

- $d$ and $n$ are, respectively, the dimension and the size of the support array
the ca will work on; hence, there are \( n^d \) finite-state automata (i.e. cells) located on the \( d \)-dimensional lattice having periodic structure (i.e. \( \mathbb{Z}^d_n \));
- \( Q \) is the finite set of cell states;
- \( N \) is the set of the \( \mathbb{Z}^d_n \)-vectors determining the neighborhood; for any \( i \in \mathbb{Z}^d_n \), if \( j \in N \) then the cell at position \( i + j \) is a neighbor of cell at position \( i \) (in what follows we shall identify the cell with its position);
- \( f : Q^{|N|} \rightarrow Q \) is the local function.

A configuration of \( \mathcal{A} \) is any element of the set \( \Sigma = Q^{\mathbb{Z}^d_n} \). Since the evolution of the system is synchronous the local map consisting of the pair \( (N, f) \) uniquely determines a global function \( F : \Sigma \rightarrow \Sigma \) in the following way; let \( N = \{j_1, \ldots, j_{|N|}\} \), then for any \( X \in \Sigma \) and \( i \in \mathbb{Z}^d_n \) the \( i \)-th component of \( F \) is determined by:

\[
F_i(X) = f(X_{i+j_1}, \ldots, X_{i+j_{|N|}}).
\]

We also denote as \( X^t \) the configuration at time \( t \geq 0 \) of the \( \mathcal{A} \)-orbit having \( X \) as the initial configuration (i.e. \( X^t = F^t(X) \)).

**Definition 1.1** A ca is weakly predictable if, for any \( X \in \Sigma \) and \( t \geq 0 \), \( F^t(X) \) is computable in \( O((\log t)n^k) \) time (where \( k \) is a fixed constant). Moreover, we call INV the class of weakly predictable ca having invertible global functions.

The State Transition Diagram of a ca \( \mathcal{A} \) is the digraph \( \mathcal{D} = (V, E) \) where \( V = \Sigma \) and \( (X, Y) \in E \) if \( Y = F(X) \). Observe that \( \mathcal{D} \) is in general not connected and each vertex has outdegree one; since the number \( |V| \) of possible configurations is finite, any system orbit, after a transient must always reach a cycle; however, the length of both transient and cyclic parts of any orbit can be exponential in \( n \). Hence, \( \mathcal{D} \) consists of cycles (which may be points) with trees rooted on cycle vertices. If a configuration \( X \in \Sigma \) belongs to a cycle then it is possible to define the value \( \text{ord}(X) \) as the least positive integer \( t \) such that \( F^t(X) = X \) (note that \( \text{ord}(X) \leq |Q|^{n^d} \) for any \( X \) in a cycle).

An important example of weakly predictable ca are those having additive local maps \([5, 9, 11]\). A ca \( \mathcal{A} = (d, n, Q, N, f) \) is additive if \( Q = (\mathbb{Z}_p, +, \cdot) \) (where \( p \) is any prime number) and, for any \( i \in \mathbb{Z}^d_n \), the local function is:

\[
X_i^t = \sum_{j \in N} (X_{i+j}^{t-1} \cdot C_{i,k}) \text{ with } C_{i,k} \in \mathbb{Z}_p
\]

(no restrictions for \( d, n \) and \( N \) are considered). Writing the above equation in vectorial form, we obtain:

\[
X_{t+1} = CX^t \quad X^t = C^tX^0 \quad t \geq 0 \quad (1)
\]

where \( C \in (\mathbb{Z}_p)^{n \times n} \) and \( C_{i,j} = 0 \) if \( j \) is not a neighbor of \( i \). Hence, the complete description of an additive ca can be given by defining a matrix \( C \in (\mathbb{Z}_p)^{n \times n} \).

Let us now introduce the interactive proof model (see [1] and [7]) and some of its important properties. The key idea is in the notion of “proof” as an interaction between two players: a Prover and a Verifier. Players interact by sending messages to each other and
after a polynomial number of interactions (in the size of input), Verifier decides whether He is “convinced” that a certain input $x$ belongs to a language $L$. Thus, Verifier is a probabilistic Turing machine working in polynomial-time with respect to the input length. On the other hand, Prover is a computationally unlimited (both in time and space) deterministic Turing machine. The length of messages exchanged between the machines is bounded by a suitable polynomial in the input length. Finally, Verifier can, during its turn, terminate the interactive computation by entering in a final (accepting or rejecting) state. The acceptance criterion is equivalent to that of $BPP$-machines:

**Definition 1.2** A language $L$ has an interactive protocol if, for any $x \in L$, there is a Prover such that Verifier accepts $x$ with probability greater than $2/3$. On the contrary, if $x \not\in L$, Verifier accepts $x$ with probability smaller than $1/3$ for any Prover. Furthermore, we define $AM[k(n)]$ the class of all languages admitting an interactive protocol using at most $k(|x|)$ interactions (rounds) between players for any input $x$.

**Theorem 1.1** [1] $AM[k] \equiv AM[2]$ for any fixed constant $k$, and $AM[2] \subset PH$ where $PH$ is the polynomial-time hierarchy (more precisely, $AM[2] \subset II^P_2$).

**Theorem 1.2** [2] If $CoNP$ is contained in $AM[k]$ for some constant $k$, then $PH$ is contained in $AM[k]$, that is $PH$ collapses.

## 2 An Interactive Protocol for CREP

Given any ca $A \in INV$ with invertible global function $F$ we can verify that any of its configurations belongs to a cycle:

**Lemma 2.1** Let $X$ and $Y$ be two arbitrary configurations; if a positive integer $t$ such that $F^t(X) = Y$ exists, then a positive integer $t' \leq |Q|^n$ such that $F^t'(Y) = X$ exists.

**Proof.** The existence of such a $t'$ is a direct consequence of the invertibility of $F$ and the resulting fact that every configuration has exactly one predecessor. □

This lemma implies that the relation in the space $\Sigma$, defined as $X \equiv_F Y$ if an element $t \in \{1, \ldots, |Q|^n\}$ exists such that $F^t(X) = Y$, is an equivalence one. Thus, the state transition diagram of a ca belonging to INV consists of a set of disconnected cycles corresponding to the partition induced by the above equivalence relation. Then CREP, for the class INV, is equivalent to, given a ca $A$ with global function $F$ and two configurations $X$ and $Y$, decide whether $X \equiv_F Y$.

Now, we can give the interactive protocol for CREP’s complement (for brevity’s sake, given a finite set $S$, we shall use the notation $h \in \alpha S$ to denote the random choice of the element $h$ according to the uniform distribution defined on $S$):
Protocol A1:

- **Input**: $A \in \text{INV}$ and the pair $(X_1, X_2) \in \Sigma \times \Sigma$;
- **Verifier** chooses $i \in \{1, 2\}$ and $t \in \{1, \ldots, |Q|^n\}$; moreover, he computes $F^t(X_i) = Z$;
- **Verifier** sends $Z$ to **Prover** and asks (keeping both $i$ and $t$ secret) to him whether $X_1 \equiv_F Z$ or $X_2 \equiv_F Z$;
- **Prover** replies to **Verifier** by sending the index $i'$;
- **Verifier** checks whether $i'$ is correct (i.e. if $i' = i$);
- **Verifier** will accept the input if and only if $i'$ is correct.

**Theorem 2.1** **CREP** for the class **INV** is in **CoAM**[2]. Thus, for this class, **CREP** cannot be NP-complete unless the polynomial-time hierarchy **PH** collapses.

**Proof.** From theorem 1.1 and theorem 1.2, it will be sufficient to prove that with a constant number of independent runs of protocol A1 on the same input (observe that A1 has only two interactions between players) we can satisfy the conditions given in definition 1.2. All computations done by **Verifier** in this protocol can be realized in polynomial time by a probabilistic Turing machine since $A$ is weakly predictable and, moreover, the message length is bounded by $n$.

Let us first suppose that $X_1 \not\equiv_F X_2$; then **Prover** can find the correct index $i \in \{1, 2\}$ such that $X_i \equiv_F Z$ since this is unique. Thus, **Verifier** will accept with probability equal to one.

Conversely, suppose that $X_1 \equiv_F X_2$ (so we have $\text{ord} = \text{ord}(X_1) = \text{ord}(X_2)$) and consider the following pair of sets:

$$ P^*_2 = \{ t \in \{1, \ldots, |Q|^n\} : F^t(X_i) = Z \} $$

where $i = 1, 2$. It is not hard to prove that $\Delta = |P^*_2| - |P^*_2|$ $\leq 1$ for any **CREP**’s instance and so only two cases may arise:

(a) if $\Delta = 0$ (this, for example, happens when $\text{ord}$ is a factor of $|Q|^n$) then **Prover** can only choose uniformly at random $i'$ (i.e. $i' \in \{1, 2\}$) since He may not “see” both $t$ and $i$ in the protocol. Hence, in this case, the probability that **Prover** gives the correct answer is $1/2$.

(b) If $\Delta = 1$ then the best strategy for **Prover** is to choose $i'$ such that $|P^*_1| = \max\{|P^*_1|, |P^*_2|\}$. Without loss of generality, we can suppose that $i' = 1$, so by considering the following definitions:

$$ Pr(E_i) = Pr_{t \in \{1, \ldots, |Q|^n\}}(F^t(X_i) = Z), $$

where $i = 1, 2$, we obtain:

$$ Pr(E_1) = \frac{\alpha + 1}{|Q|^n} \text{ and } Pr(E_2) = \frac{\alpha}{|Q|^n} $$

where $\alpha = [|Q|^n/\text{ord}]$ (observe that $\alpha \geq 1$ for any input). Thus, in this case, the probability that **Prover** will guess the correct index (i.e. $i' = i$) is (omitting some simple computation):

$$ Pr(E_1 | E_1 \lor E_2) = \frac{\alpha}{2\alpha + 1} + \frac{1}{2\alpha + 1} \leq \frac{5}{6} $$
Finally, let us consider the global protocol consisting of seven independent runs of $A_1$ on the same input and where we let Verifier accept the input if and only if Prover gives the correct index $i'$ for each of these seven runs. Thus in both $a)$ and $b)$ cases, if $X_1 \equiv_F X_2$ the probability that Verifier accepts is not greater than $(5/6)^7 < 1/3$ and the theorem is completely proved (observe that, in the first case two independent runs of $A_1$ would be sufficient).

When the weakly predictable $ca$ is not invertible, protocol $A_1$ cannot be applied. However, if in a particular $ca$ the transient length of any orbit is polynomially bounded by $n$ and, furthermore, it is possible to detect whether a given configuration is (or not) in a cycle, then Protocol $A_1$ will work correctly. Every additive $ca$ satisfies the above conditions, indeed using basic results in linear algebra it is possible to prove the following facts:

**Lemma 2.2** [9] Let $A$ be an additive $ca$; $A$ is invertible if and only if its matrix $C$ (see equation 1) has no eigenvalue $0$, that is, if and only if $\text{det}(C) \neq 0$. Suppose $A$ is not invertible, then $X$ belongs to a cycle if and only if it has no components on the eigenspace $S_0$ determined by the eigenvalue $\lambda = 0$ of $C$ (this fact can be decided by computing the Jordan form of $C$). Finally, if $X$ is not in a cycle the transient length $l$ of its orbit is bounded by the dimension of $S_0$, thus $l \leq n$.

Finally, by using the above lemma, we have:

**Theorem 2.2** CREP for additive cellular automata is in CoAM[2]. Thus, CREP cannot be NP-complete unless the polynomial-time hierarchy collapses.

*Proof.* From lemma 2.2, given any CREP’s instance $\langle A, X, Y \rangle$ in which $A$ is an additive $ca$, three cases may arise (observe that it is possible to detect in polynomial time in which of these cases $\langle A, X, Y \rangle$ belongs to):

i) $X$ and $Y$ are both in the transient of some orbit; then, by using the last fact of lemma 2.2, we can decide $\langle A, X, Y \rangle$ in polynomial time.

ii) $X$ is in a cycle and $Y$ is in a transient of some orbit; then we can certainly state that $\langle A, X, Y \rangle$ is a NO-instance of CREP.

iii) $Y$ is in a cycle; then $\langle A, X, Y \rangle$ is polynomial-time reducible to the instance $\langle A, X^c, Y \rangle$ where $X^c$ is the first configuration belonging to the cyclic part of the $A$-orbit starting from $X$; indeed, it is possible to compute $X^c$ in polynomial time since the the transient length of any orbit is not greater than $n$. Finally, we can apply protocol $A_1$ on this new instance (also in this case, the correctness of $A_1$ can be proved in the same way of that of theorem 2.1). □

6


References


