Introducing Freezing Cellular Automata*

Eric Goles[1], Nicolas Ollinger[2], and Guillaume Theyssier[3]

[1] Facultad de Ingeniería y Ciencias, Universidad Adolfo Ibañez, Santiago, Chile
[2] Univ. Orléans, INSA Centre Val de Loire, LIFO EA 4022, FR-45067 Orléans, France
[3] CNRS, Université de Savoie (LAMA, UMR 5127), France

Abstract. We introduce the class of freezing cellular automata (CA): those where the state of a cell can only increase according to some order on states. It contains some well-studied examples like the bootstrap percolation CA or "life without death", but here we aim at studying the class as a whole and deriving general properties of freezing CA. Our focus is mainly on the complexity of these CA and we show that, if their definition imposes strong constraints on their possible dynamics, they still can produce complex computational behaviors, even in dimension 1. Our main results are that the prediction problem of these CA can be P-complete in dimension 2 or more, but is always NL in dimension 1. Moreover its communication complexity is always at most $O(n^{d-1}\log(n))$ in dimension d (while it can be $\Omega(n^d)$ for a CA in general). As another dimension-sensitive property, we show that the nilpotency problem is decidable in dimension 1 but not in higher dimension. Finally, although simpler, we show that one-dimensional freezing CA can still be Turing universal.

1 Definition and Examples

A cellular automaton of dimension d and state set Q is a map F acting on the set of configuration $Q^{\mathbb{Z}^d}$ in a continuous and translation invariant way (see [5]). Equivalently, it can be define locally by a neighborhood V (a finite subset of $\mathbb{Z}^d$) and a local transition function $f : Q^V \to Q$ as follows:

$$\forall z \in \mathbb{Z}^d, \quad F(c)_z = f\big(c|_{z+V}\big) \quad .$$

In this paper we focus on a particular class of CA defined by a simple condition on its local transition function as follows.

Definition 1. *A CA F is a* freezing CA*, for some (partial) order $\geq$ on states, if the state of any cell can only increase, i.e.*

$$F(c)_z \geq c_z$$

for any configuration c and any cell z.

* This work has been supported by ECOS Sud – CONICYT project C12E05

In a freezing CA, a cell can change at most a finite number of time during its evolutions, precisely at most $n-1$ times for a CA with n states. Conversely, a CA with a uniformly bounded number of state change at each cell is not necessarily freezing. For instance, the following nilpotent CA over $\{0, 1, 2\}$ is not freezing:

$$f(a, b) = \begin{cases} 1 & \text{if } b = 0 \\ 2 & \text{else.} \end{cases}$$

because both $2 \to 1$ and $1 \to 2$ are possible state changes in one cell (but after 2 steps all cells are in state 2).

However, it is easy to simulate any CA with state set Q and at most k changes per cell by a freezing CA on $Q \times \{0, \ldots, k\}$: the $\{0, \ldots, k\}$ component is increased each time the original CA generate a state change in the Q component, and when the value k is reached, any state change is prohibited. The resulting CA is freezing for the order $\preceq$ on states $Q \times \{0, \ldots, k\}$ defined by:

$$(q, i) \preceq (q', j) \Leftrightarrow q = q' \wedge i \leq j$$

CA with a bounded number of state change have been as language recognizer [14] (see also [9] for CA with bounded communications). Here we study freezing CA as dynamical systems.

Particular freezing cellular automata have already been considered in the literature, in particular the threshold growth models in 2D [3]: they are CA with states $0, 1$ where 0 becomes 1 if the number of 1s in the neighborhood is above some threshold, and 1s stay unchanged forever. They were in particular considered as theoretical models of bootstrap percolation and a lot of work was dedicated to the experimental and rigorous analysis of the phase transitions they exhibit (see for instance [6]).

Another freezing 2D CA with 2 states called "life without death" receive a lot of attention and was in particular shown P-complete [4].

Finally, other well-studied models which are not exactly deterministic cellular automata but of similar kind fulfill the freezing paradigm, for instance:

- self-assembly tilings [12]: tiles are added in empty places but never removed;
- SIR epidemic propagation models [1]: a **S**usceptible person can become **I**nfected, and then **R**ecover and acquire immunity so that it nether goes back to susceptible or infected states.

2 Basic Dynamical Properties

The family of freezing CA is closed under iterations, Cartesian product, subautomaton, local factor and grouping, but not under composition by shifts: the identity automaton is freezing, but the shift automaton is not.

The main property of these CA is that they progressively "freeze" any initial configuration.

Proposition 1. *Given a freezing CA F and any configuration c, the sequence $\left(F^t(c)\right)_t$ converges in the Cantor metric to some limit configuration $F^\infty(c)$ which is a fixed point for F.*

Proof. Since the state of any cell can only increase, each cell can only change its state a finite number of times. Therefore, starting from configuration x, for any $n \geq 0$ there exists a time t such that no cell at distance n from the center will change of state after time t. This is equivalent to the convergence of the sequence $\left(F^t(x)\right)_t$. ■

This "freezing" behavior is intrinsically irreversible except in the trivial case of the identity.

Proposition 2. *If a freezing CA is surjective then it is the identity map.*

Proof (Proof sketch). Suppose that F is a freeezing CA which is not the identity map. Then there must exist two configurations c and d such that $F(c) = d$ but $c_0 \neq d_0$. Let's choose such a pair (c, d) that minimizes (for the order $\leq$ on states) the state d_0. Because F is freezing, we have $c_0 \leq d_0$. Having chosen d_0 minimal, this implies that for any c', d' with $F(c') = d'$ and $d'_0 = c_0$, we must also have $c'_0 = c_0$.

Therefore, state c_0 can only be produced from state c_0 but in at least one context (appearing in configuration c), c_0 changes to another state (d_0): the CA F cannot be balanced and therefore is not surjective [10]. ■

Proposition 1 says that every orbit converges to a fixed point. This does not imply that the limit set is made of fixed points, however we can prove the following proposition which is not true for CA in general.

Proposition 3. *Let F be a freezing CA which is not nilpotent. Then it possesses two distinct fixed points.*

Proof. Let M be a maximal state for the order $\leq$ on states coming from the freezingness of F, and denote by c_M the uniform configuration everywhere equal to M. Since F is freezing, c_M must be a fixed point for F. Now, since F is not nilpotent, for any $t \geq 0$, there is some configuration c_t such that $F^t(c_t)_0 \neq M$. Because F is freezing, for any fixed $k \geq 0$, the value of the cells at distance k of the center can change at most $\phi(k)$ times in any orbit. Therefore choosing $t > \phi(k)$, there must be some step $t' \leq t$ in the orbit of c_t at which these cells do not change, formally:

$$\forall i, \quad \|i\| \leq k \Rightarrow F^{t'}(c_t)_i = F^{t'+1}(c_t)_i.$$

Denote by $d_k = F^{t'}(c_t)$. For any k we have:

- $\forall i, \|i\| \leq k, F(d_k)(i) = d_k(i)$
- $d_k(0) \neq M$ (because $F^t(c_t)_0 \neq M$, F is freezing and M is a maximal state).

By compactness, we can extract a limit point d from the sequence $(d_k)_k$. Finally, by the properties of the d_k stated above, we have: $F(d) = d$ and $d \neq c_M$. ■

As a corollary, we get the following dimension-sensitive decidability result.

Theorem 1. *The nilpotency problem for freezing CA is:*

- *decidable (in polynomial time) for 1D CA;*
- *undecidable for CA in higher dimension.*

Proof. First, in dimension 2 and more, the classical proof of undecidability of nilpotency (see [7]) works without any modification for freezing CA: given a Wang tile set, we build a freezing CA with a spreading error state, that checks locally if the configuration is a valid tiling and produces the error state in case of local error detection. This CA is nilpotent if and only if the tile set does not tile the plane.

In dimension 1, from Proposition 3, we know that nilpotency is equivalent to the following first-order property

$$\forall x \forall y \big(x = F(x) \wedge y = F(y)\big) \Rightarrow x = y\big).$$

From [13], it follows that it is decidable. ■

3 Computational Complexity

Definition 2. *Let F be any CA of radius r and dimension d. The prediction problem* PRED_F *is defined as follows:*

- *input: a square pattern u of size $(2rt+1)^d$*
- *ouput: the state $F^t(u)$*

Proposition 4. *2D freezing CA are Turing-universal and there exists a 2D freezing CA with a P-complete prediction problem.*

Proof. Any 1D CA F with states Q and neighborhood V can be simulated by a 2D freezing CA with states $Q \cup \{*\}$ as follow. Let $V' = \{(v, -1) : v \in V\}$. A cell in a state from Q never changes. A cell in states $*$ look at cells in its V' neighborhood: if they are all in a state from Q then it updates to the state given by applying F on them, otherwise it stays $*$. Starting from a all-$*$ configuration except on one horizontal line where it is a Q-configuration c_0, this 2D freezing CA will compute step by step the space-time diagram of F on configuration c_0. The proposition follows. ■

The following result for freezing CA was present in [14] in a different formalism for bounded-change CA.

Proposition 5. *The prediction problem of any 1D freezing CA is NLOGSPACE.*

Proof (Proof sketch). Consider (without loss of generality) a 1D freezing CA F of radius 1 with k states. To know that F gives state q after n steps on some input u of size $2n+1$ it is sufficient to guess a sequence of $2n+1$ columns of states $(C_i)_{-n\leq i\leq n}$, where C_i is of height $n-|i|+1$, and to check that they form a (triangular) valid space-time diagram of F on input u that leads to state q (last letter of C_0).

The key observation is that, since F is freezing, at most k changes of state can occur in any column, hence a column of height n can be represented in space $O(\log(n))$ (constant list of changes, each given by the time step at which it occurs and the new state).

We give the non-deterministic LOGSPACE algortihm to solve the prediction problem of F:

- **initialization:** guess columns C_{-n} and C_{-n+1} and set current position to $-n+1$;
- **main loop:** if at current position p, guess column C_{p+1} with the condition $C_{p+1}[0]=u_{p+1}$ and check that each position in C_p is compatible with the neighboring columns C_{p-1} and C_{p+1}, formally, for all i with $0\leq i<|C_p|$ check that $f(C_{p-1}[i],C_p[i],C_{p+1}[i])=C_p[i+1]$ where f is the local function of F; increment current position;
- **termination:** if current position is $n+1$ return $C_0[n]$

Note that the compatibility test between neighboring columns in the main loop can be done using the compact log-space representation of columns. ■

Theorem 2. *1D freezing CA are Turing-universal.*

Proof. Any k-counter Minsky machine [11] with states Q can be simulated by a 1D freezing CA encoding the evolution of both the state and counter values on signals moving through the space-time diagram, one transition per space unit. The control of the machine is represented as a signal of equation $0\leqslant y-2x\leqslant 1$ where both cells in column x carry the state of the counter machine and an emptiness flag for each counter at time x. Each of the k counters is represented as a signal of the same slope so that the distance to the control in column x encodes the value of the counter at time x. A counter can easily be tested for equality to zero. Incrementation corresponds to a local deceleration of the signal and decrementation to a local acceleration. Incrementation, decrementation or no-operations decisions are carried from the control to the counter as a simple vertical signal. Such a simulation can be implemented by a CA with neighborhood $\{-1,0\}$ and $1+8|Q|+(3+3+1)^k$ states, as depicted on Fig. 1. The Turing-universality is then a consequence of the universality of Minsky machines.

To see that the constructed CA is indeed freezing, let us consider a typical column in the simulation: blank symbols are followed by the two control states, followed by k independent counter column. Each counter column contains an operation followed by one to three counter encoding followed by a new blank symbol. A partial order is naturally derived from there where the independent counter columns are treated by a product order. ■

2++w	--	--	--	--	--	--	--	--	--	--	--	--	--
2++	--	--	--	--	--	--	--	--	--	--	--	--	--
(+-)	2++w	--	--	--	--	--	--	--	--	--	--	--	--
(+-)	2++	--	--	--	--	--	--	--	--	--	--	--	--
(+-)	(+-)	2++w	--	--	--	--	--	--	--	--	--	--	--
(+-)	(+-)	2++	--	--	--	--	--	--	--	--	--	--	--
(k-)	(+-)	(+-)	2++w	--	--	--	--	--	--	--	--	--	--
(C-)	(+-)	(+-)	2++	--	--	--	--	--	--	--	--	--	--
(c-)	(+-)	(+-)	(+-)	2++w	--	--	--	--	--	--	--	--	--
(.c)	(k-)	(+-)	(+-)	2++	--	--	--	--	--	--	--	--	--
(..)	(Cc)	(+-)	(+-)	(+-)	2++w	--	--	--	--	--	--	--	--
(..)	(c.)	(+c)	(+-)	(+-)	2++	--	--	--	--	--	--	--	--
(..)	(..)	(k.)	(+c)	(+-)	(+-)	2++w	--	--	--	--	--	--	--
(..)	(..)	(C.)	(+.)	(+c)	(+-)	2++	--	--	--	--	--	--	--
(..)	(..)	(c.)	(+.)	(+.)	(+c)	(+-)	2++w	--	--	--	--	--	--
(..)	(..)	(..)	(k.)	(+.)	(+.)	(+c)	2++	--	--	--	--	--	--
(..)	(..)	(..)	(C.)	(+.)	(+.)	(+.)	(+c)	2++w	--	--	--	--	--
(..)	(..)	(..)	(c.)	(+.)	(+.)	(+.)	(+.)	2+z	--	--	--	--	--
(..)	(..)	(..)	(..)	(k.)	(+.)	(+.)	(+.)	(.)	0+zw	--	--	--	--
(..)	(..)	(..)	(..)	(C.)	(+.)	(+.)	(+.)	(.)	0+z	--	--	--	--
(..)	(..)	(..)	(..)	(c.)	(+.)	(+.)	(+.)	(.)	(-.)	1+zw	--	--	--
(..)	(..)	(..)	(..)	(..)	(k.)	(+.)	(+.)	(.)	(-.)	1+z	--	--	--
(..)	(..)	(..)	(..)	(..)	(C.)	(+.)	(+.)	(.)	(-.)	(-C)	0++w	--	--
(..)	(..)	(..)	(..)	(..)	(c.)	(+.)	(+.)	(.)	(-.)	(-c)	0++	--	--
(..)	(..)	(..)	(..)	(..)	(..)	(k.)	(+.)	(.)	(-.)	(-.)	(-C)	1++w	--
(..)	(..)	(..)	(..)	(..)	(..)	(C.)	(+.)	(.)	(-.)	(-.)	(-c)	1++	--
(..)	(..)	(..)	(..)	(..)	(..)	(c.)	(+.)	(.)	(-.)	(-.)	(-.)	(-k)	0++w
(..)	(..)	(..)	(..)	(..)	(..)	(..)	(k.)	(.)	(-.)	(-.)	(-.)	(-C)	0++
(..)	(..)	(..)	(..)	(..)	(..)	(..)	(C.)	(.)	(-.)	(-.)	(-.)	(-c)	(-)
(..)	(..)	(..)	(..)	(..)	(..)	(..)	(c.)	(.)	(-.)	(-.)	(-.)	(-.)	(-C)
(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(C.)	(-.)	(-.)	(-.)	(-.)	(-c)
(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(c.)	(-.)	(-.)	(-.)	(-.)	(-.)
(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(c.)	(-.)	(-.)	(-.)	(-.)
(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(c.)	(-.)	(-.)	(-.)
(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(c.)	(-.)	(-.)
(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(c.)	(-.)
(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(c.)
(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)
(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)
(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)	(..)

Fig. 1. A 1D freezing CA simulating a 2-CM

4 Communication complexity

Communication complexity was introduced by Yao [15] to study distributed computing. We first briefly recall the classical definition of communication complexity for any function (see [8] for general reference), and then apply it to the prediction problem of cellular automata (for more details see [2]). The goal of this section is to show that freezing CA have a lower communication complexity than CA in general.

Consider a function $f : A \times B \to C$ where A, B and C are finite sets. The communication complexity of f is the amount of information that need to be exchanged in the worst case between two parties (say Alice and Bob) in order to decide the value $f(x, y)$ when one of the parties knows only x and the other only y (both know f and they have unlimited computing power). Intuitively, the communication complexity of f is defined as the cost of the best protocol that solves f. A protocol is an agreement between the two parties on how to communicate by rounds until one knows the correct value $f(x, y)$. The cost of a protocol is the total amount of information (in bits) exchanged in the worst case over all inputs. We refer to [8, 2] for more precise definitions.

For any CA F we consider the communication complexity of the prediction problem by dividing the space into two parts through an hyperplane: given some i, Alice has access to all cells whose first coordinate is less than i and Bob to all the others. The upper-bound that we give does not depend on this choice of i.

Theorem 3. *For any freezing CA F of dimension d the communication complexity of* PRED_F *on inputs of size n^d is $O\big(n^{d-1}\log(n)\big)$.*

Proof (Proof sketch). Let r be the radius of F. The cells that Alice and Bob respectively have access to are separated by an hyperplane at position i. We distinguish two zones of the space that play an important role: the *A-zone* is the set of cells (known by Alice) between the hyperplane at position $i - r$ and the one at position i, the *B-zone* is the set of cells (known by Bob) between hyperplane i and $i + r$. The general idea is that Alice (resp. Bob) only needs to know the content of the B-zone (resp. A-zone) in addition to its own cells in order to correctly compute one step of F on its part of the space. The protocol is as follows: first, as an initialization step, Alice and Bob exchange the content of the A and B-zones and initialize their time counter at 0. Then, they repeat as much as necessary the following loop:

1. Alice (resp. Bob) separately compute the evolution step by step on its own half-space **supposing that nothing changes** in the B-zone (resp. A-zone) and **until a state change is observed** in the A-zone (resp. B-zone);
2. Alice (resp. Bob) send a "diff report" which contains the time step reached and the list of cell changes that occurred in its A-zone (resp. B-zone);
3. Alice (and symmetrically for Bob) receives the other "diff report" and updates its knowledge as follows:
   - sets its time counter as the min t_m of the time values present in the two reports;

  - if the time in the received diff report is strictly larger than her own time, then ignore the report and keep its previous knowledge of the B-zone, otherwise updates her knowledge of the B-zone according to the diff report;
  - if the time in the received diff report is strictly smaller than her own time, then revert its knowledge of her half-space to what it was at time t_m (which she can do without any further information from Bob).

This protocol allows Alice and Bob to correctly compute the evolution of F as far as they need (without even supposing that it is freezing) and thus allows to solve the prediction problem of F. Indeed, the fact that they retrospectively jump back in time to the step of the first change in the A/B-zones ensures that the hypothesis of no change in the A/B-zone is correct for all the retained computation steps.

Let's now upper bound its cost on an input of size n^d:

- the initialization costs $O(n^{d-1})$ (the sizes of the A-zone and the B-zone restricted to the input are n^{d-1});
- each cell change notification in the diff reports costs at most $O(\log(n))$ ($\log(n)$ for the time step, $O(\log(n))$ for communicating the position of the cell, a constant to describe the new state);
- since F is freezing, the total number of changes in the A-zone and B-zone is $O(n^{d-1})$ (each cell can change a constant number of times).

Therefore the total cost of the protocol is at most $O\big(n^{d-1}\log(n)\big)$. ■

References

1. Fuentes, M., Kuperman, M.: Cellular automata and epidemiological models with spatial dependence. Physica A: Statistical Mechanics and its Applications 267(3–4), 471 – 486 (1999)
2. Goles Ch., E., Meunier, P., Rapaport, I., Theyssier, G.: Communication complexity and intrinsic universality in cellular automata. Theor. Comput. Sci. 412(1-2), 2–21 (2011)
3. Gravner, J., Griffeath, D.: Cellular automaton growth on z2: Theorems, examples, and problems. Advances in Applied Mathematics 21(2), 241 – 304 (1998)
4. Griffeath, D., Moore, C.: Life without death is P-complete. Complex Systems 10 (1996)
5. Hedlund, G.A.: Endomorphisms and Automorphisms of the Shift Dynamical Systems. Mathematical Systems Theory 3(4), 320–375 (1969)
6. Holroyd, A.E.: Sharp metastability threshold for two-dimensional bootstrap percolation. Probability Theory and Related Fields 125(2), 195–224 (2003)
7. Kari, J.: The Nilpotency Problem of One-dimensional Cellular Automata. SIAM Journal on Computing 21, 571–586 (1992)
8. Kushilevitz, E., Nisan, N.: Communication complexity. Cambridge university press (1997)
9. Kutrib, M., Malcher, A.: Cellular automata with sparse communication. Theor. Comput. Sci. 411(38-39), 3516–3526 (2010)

10. Maruoka, A., Kimura, M.: Condition for Injectivity of Global Maps for Tessellation Automata. Information and Control 32, 158–162 (1976)
11. Minsky, M.: Computation: Finite and Infinite Machines. Prentice Hall, Englewoods Cliffs (1967)
12. Patitz, M.J.: An introduction to tile-based self-assembly and a survey of recent results. Natural Computing 13(2), 195–224 (2014)
13. Sutner, K.: Model checking one-dimensional cellular automata. J. Cellular Automata 4(3), 213–224 (2009)
14. Vollmar, R.: On cellular automata with a finite number of state changes. In: Knödel, W., Schneider, H. (eds.) Parallel Processes and Related Automata / Parallele Prozesse und damit zusammenhängende Automaten, Computing Supplementum, vol. 3, pp. 181–191. Springer Vienna (1981)
15. Yao, A.C.C.: Some complexity questions related to distributive computing (preliminary report). In: STOC. pp. 209–213 (1979)