Permutive One-Way Cellular Automata and the Finiteness Problem for Automaton Groups

Martin Delacourt(✉) and Nicolas Ollinger

Univ. Orléans, LIFO EA 4022, 45067 Orléans, France
{martin.delacourt,nicolas.ollinger}@univ-orleans.fr

Abstract. The decidability of the finiteness problem for automaton groups is a well-studied open question on Mealy automata. We connect this question of algebraic nature to the periodicity problem of one-way cellular automata, a dynamical question known to be undecidable in the general case. We provide a first undecidability result on the dynamics of one-way permutive cellular automata, arguing in favor of the undecidability of the finiteness problem for reset Mealy automata.

Keywords: Reset Mealy automata · One-sided cellular automata · Permutive cellular automata · Periodicity problem · Reversible computation

1 Introduction

Finite-state automata provide a convenient finite description for different kinds of behavior generated by their computations. As such, Mealy automata [10] provide a finite description for the family of automaton (semi)groups that has proven its usefulness to generate interesting counter examples in the field of group theory [3]. Several decision problems inspired by algebraic questions have been studied on automaton groups: the word problem is decidable whereas the conjugacy problem is undecidable [16]. The general case of the finiteness problem remains open [1] although special cases have been solved: Gillibert [9] proved that the problem is undecidable for semigroups and Klimann [15] that it is decidable for reversible Mealy automata with two states. The status of the finiteness problem remains open for the class of reset Mealy automata.

Cellular automata [14] provide a finite description for a family of discrete dynamical systems, the endomorphisms of the shift dynamical system [11]. Decision problems inspired by dynamical questions have been investigated on cellular automata since the work of Amoroso and Patt [2]. The computation nature of cellular automata lead to sophisticated construction techniques to establish the undecidability of various decision problems like the nilpotency problem [13] or more recently the periodicity problem [12]. The status of the periodicity problem remains open for one-way cellular automata.

Without much surprise, cellular automata can be a valuable tool to establish undecidability results on Mealy automata. Indeed, Gillibert's result is inspired by Kari's proof of the undecidability of the nilpotency problem.

J. Kari et al. (Eds.): CiE 2017, LNCS 10307, pp. 234–245, 2017.
DOI: 10.1007/978-3-319-58741-7_23

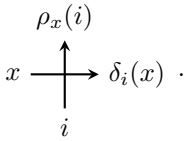
In this paper, we study the computational power of reversible permutive one-way cellular automata [4,7]. Our first contribution is a precise formalization of the connection between both open problems: the finiteness problem for reset Mealy automata is decidable if and only if the periodicity problem for one-way cellular automata is decidable. Our second contribution is a technique to embed computation inside reversible one-way cellular automata using permutive automata. The technique is applied to prove a first undecidability result on these objects.

2 Definitions

For a detailed introduction on Mealy automata, the reader is referred to Bartholdi and Silva [3] and to Kari [14] for cellular automata.

2.1 Mealy Automata

A *Mealy automaton* is a deterministic complete 1-to-1 transducer $(A, \Sigma, \delta, \rho)$, where A is a finite set of states, Σ a finite alphabet, $\delta = (\delta_i : A \to A)_{i\in\Sigma}$ is the set of transition functions and $\rho = (\rho_x : \Sigma \to \Sigma)_{x\in A}$ the set of production functions. The transition $x \xrightarrow{i|\rho_x(i)} \delta_i(x)$ is depicted by



The *production functions* naturally extend to functions on the set of finite words: $\rho = (\rho_x : \Sigma^* \to \Sigma^*)_{x\in A}$ with $\rho_x(au) = \rho_x(a)\rho_{\delta_a(x)}(u)$. The semigroup generated by the automaton is the set of all compositions of the production functions $H = \langle \rho_x : x \in A \rangle$. An *automaton semigroup* is a semigroup generated by a Mealy automaton.

A Mealy automaton is *invertible* if ρ_x is a permutation of Σ for every $x \in A$. Note that it implies that every ρ_x is also a permutation on Σ^k for every $k \in \mathbb{N}$. The group generated by a invertible Mealy automaton is $G = \langle \rho_x, \rho_x^{-1} : x \in A \rangle$. An *automaton group* is a group generated by a Mealy automaton. An invertible Mealy automaton generates a finite group if and only if it generates a finite semigroup [1].

Finiteness problem. *Given an invertible Mealy automaton, decide if the generated group is finite.*

A Mealy automaton is *reset* if, for each transition $x \xrightarrow{i|\rho_x(i)} \delta_i(x)$, the output state $\delta_i(x)$ depends only on the input letter i and not on the input state x, that is $\delta_i(x) = f(i)$ for some letter-to-state map $f : \Sigma \to A$. To simplify notations, such an automaton will be denoted as (A, Σ, f, ρ).

When studying the decidability of the finiteness problem restricted to reset automata, one can focus on the case $\Sigma = A$ and $f = Id$ as stated below.

Lemma 1. *The group generated by a reset Mealy automaton (A, Σ, f, ρ) is finite if and only if it is the case for the automaton $(\Sigma, \Sigma, 1, \rho')$ with $\rho'_x(i) = \rho_{f(x)}(i)$.*

Proof. Let G be the group generated by (A, Σ, f, ρ) and H be the group generated by $(\Sigma, \Sigma, 1, \rho')$. As every generator ρ'_x of H is a generator $\rho_{f(x)}$ of G then H is a subgroup of G. A generator ρ_y of G with $y \in A \setminus f(\Sigma)$ is not a generator of H, however as y can only be an initial state of a transition, it only impacts the size of G by a factor $n!$ where n is the size of Σ. ■

2.2 Cellular Automata

A *one-way cellular automaton* (OCA) $\mathcal{F}$ is a triple (X, r, δ) where X is the finite set of states, r is the radius and $\delta : X^{r+1} \to X$ is the local rule of the OCA. A *configuration* $c \in X^{\mathbb{Z}}$ is a biinfinite word on X. The *global function* $\mathcal{F} : X^{\mathbb{Z}} \to X^{\mathbb{Z}}$ synchronously applies the local rule: $\mathcal{F}(c)_i = \delta(c_i, \dots, c_{i+r})$ for every $c \in X^{\mathbb{Z}}$ and $i \in \mathbb{Z}$. The *spacetime diagram* $\Delta : \mathbb{Z} \times \mathbb{N} \to X$ generated by an initial configuration c is obtained by iterating the global function: $\Delta(k, n) = \mathcal{F}^n(c)_k$ for every $k \in \mathbb{Z}$ and $n \in \mathbb{N}$. Following Hedlund [11] characterization of cellular automata as endomorphisms of the shift, we assimilate an OCA, with minimal radius, and its global function.

Notice that OCA are the restriction of classical cellular automata (CA) where a cell only depends on other cells on the right side. The *identity* function Id, the *left shift map* σ_l and the XOR rule are OCA, respectively encoded as $(X, 0, 1)$, $(X, 1, (x, y) \mapsto y)$ and $(\{0, 1\}, 1, \oplus)$, whereas the *right shift map* σ_r is not.

A state $x \in X$ is *quiescent* if $\delta(x, \dots, x) = x$. A configuration is *finite* if it contains the same quiescent state x everywhere but on finitely many positions.

An OCA is (left) *permutive* if the map $x \mapsto \delta(x, x_1, \dots, x_r)$ is a permutation of X for every $(x_1, \dots, x_r) \in X^r$. An OCA is *periodic* of period $T > 0$ if $\mathcal{F}^T = \mathrm{Id}$.

Periodicity problem. *Given a cellular automaton, decide if it is periodic.*

An OCA is *reversible* if its global function is bijective with an inverse that is also an OCA. A periodic OCA $\mathcal{F}$ of period T is reversible of inverse $\mathcal{F}^{T-1}$. Following Hedlund [11], the inverse of a bijective OCA is always a CA, however usually not one-sided. The following lemmas assert that this technical issue disappears by considering only permutive OCA.

Lemma 2. *Every reversible OCA is permutive.*

Proof. Let $\mathcal{F}$ be a reversible OCA. As both $\mathcal{F}$ and $\mathcal{F}^{-1}$ are OCA, $\mathcal{F}$ is bijective on $X^{\mathbb{N}}$ too. Let $(x, x', x_1, \dots, x_r) \in X^{r+2}$ and $c = x_1 \cdots x_r x_r^{\omega}$. If $x \neq x'$, as $\mathcal{F}(xc) \neq \mathcal{F}(x'c)$, we have $\delta(x, x_1, \dots, x_r) \neq \delta(x', x_1, \dots, x_r)$. ■

Lemma 3. *Every bijective permutive OCA is reversible.*

Proof. Let $\mathcal{F}$ be a bijective permutive OCA. By permittivity $f : X \times X^{\mathbb{N}} \to X$ defined by $f(y, c) = x$ where $\mathcal{F}(xc) = y\mathcal{F}(c)$ is well defined and permutive in its first argument. Let $u, u' \in X^{-\mathbb{N}}$ and $v' \in X^{\mathbb{N}}$. Let $w \in X^{-\mathbb{N}}$ and $v \in X^{\mathbb{N}}$ be such that $\mathcal{F}^{-1}(u'v') = wv$. Let $w' \in X^{-\mathbb{N}}$ be defined recursively by

$w'_i = f(u_i, w'_{i-1} \cdots w'_0 v)$. By construction $\mathcal{F}^{-1}(uv') = w'v$. As $\mathcal{F}^{-1}(uv')$ and $\mathcal{F}^{-1}(u'v')$ are equal on $\mathbb{N}$ for all u, u', v' the CA $\mathcal{F}^{-1}$ is an OCA. ■

Note that the inverse of a bijective permutive OCA can have a larger radius. However, as bijectivity is decidable for cellular automata [2] and as bijectivity is preserved by grouping cells [8], when studying the decidability of the periodicity problem restricted to permutive OCA, one can focus on the case of reversible permutive OCA with radius 1 and inverse radius 1 syntactically characterized by the following lemma (already proven in [4]).

Lemma 4. *An OCA $(X, 1, \delta)$ is reversible with inverse radius 1 if and only if it is permutive and for all $x, y, x', y' \in X$, if $\delta(x, y) = \delta(x', y')$ then $\pi_x = \pi_{x'}$ where π_y maps x to $\delta(x, y)$.*

From now on, we only consider these OCA.

3 Linking Finiteness and Periodicity

Reset Mealy automata with $\Sigma = A$ and $f = Id$ and permutive OCA of radius 1 are essentially deterministic complete letter-to-letter transducers of a same kind, as depicted on Fig. 1. The following proposition formalizes this link.

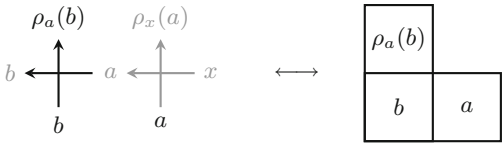


Fig. 1. Linking reset Mealy automata to permutive OCA

Proposition 1. *The group generated by a reset automaton $(\Sigma, \Sigma, 1, \rho)$ is finite if and only if the permutive OCA $(\Sigma, 1, \delta)$, where $\delta(x, y) = \rho_y(x)$, is periodic.*

Proof. First note that the following equations hold for all $u_0, u_1, \ldots, u_k \in \Sigma$:

$$\rho_{u_k}(u_{k-1}, u_{k-2}, \ldots, u_0) = \rho_{u_k}(u_{k-1})\rho_{u_{k-1}}(u_{k-2}) \ldots \rho_{u_1}(u_0)$$
$$\delta(u_0, u_1, \ldots, u_k) = \rho_{u_1}(u_0)\rho_{u_2}(u_1) \ldots \rho_{u_k}(u_{k-1})$$

By extension, for all $k > t > 0$ and for all words $u \in \Sigma^t$ and $v \in \Sigma^k$, the following equation holds: $\rho_u(v)_k = \delta^t(v_k, v_{k-1}, \ldots, v_{k-t})$.

Suppose now the group generated by the reset Mealy automaton is finite. Let $a \in \Sigma$ be any letter and let n be the order of ρ_a, then $\rho_{a^n} = (\rho_a)^n = \mathrm{Id}$ thus $\delta^n = \mathrm{Id}$, the OCA is periodic.

Conversely, let n be the period of the OCA. By previous remarks, for all $k > 0$ and words $u, v \in \Sigma^n$, $w \in \Sigma^k$, the image $\rho_u(vw)$ is $v'w$ for some $v' \in \Sigma^n$. The set of ρ_u generates a subgroup of permutations of Σ^n, which is finite, when u takes all possible values in Σ^n. The automaton group is finite. ■

Corollary 1. *The finiteness problem restricted to reset Mealy automata is decidable if and only if the periodicity problem restricted to OCA is decidable.*

Notice that the situation described by this corollary is optimal: if the problem is decidable, Mealy automata is the right setting to prove this result and the decidability of the periodicity for OCA will be a consequence; if the problem is undecidable, cellular automata is the right setting to prove this result and the undecidability of the finiteness problem will be a consequence. The remainder of this paper is dedicated to prove that computational phenomena do appear inside the dynamics of permutive OCA, advocating for the undecidability of the problem.

Conjecture 1. The finiteness problem is undecidable.

4 Computing with Permutive OCA

As shown in Fig. 1, the time goes up in every representation of this paper.

Given a permutive OCA, we show how to build a reversible OCA that can simulate every spacetime diagram of the original. The idea is to slow down the computation by delaying each state using a fixed number of distinct copies per state and perform a transition only after going through every copy. Adjacent columns of states are then desynchronized to obtain reversibility as a consequence of permittivity.

Definition 1. *Let $\mathcal{F}$ be an OCA $(X, 1, \delta)$ and let $1 \leq k \leq n-1$. The (n,k)-embedding $\mathcal{F}'$ of $\mathcal{F}$ is the OCA $(X', 1, \delta')$ where $X' = \bigcup_{1 \leq i \leq n} \{x^{(i)} : x \in X\}$ and such that:*

$$\forall x^{(\alpha)}, y^{(\beta)} \in X' \qquad \delta'\left(x^{(\alpha)}, y^{(\beta)}\right) = \begin{cases} \delta(x,y)^{(1)} & \text{if } \alpha = n \text{ and } \beta = k \\ x^{(1+(\alpha \mod n))} & \text{otherwise} \end{cases}$$

Figure 2 illustrates the embedding.

Lemma 5. *The (n,k)-embedding of a permutive OCA $\mathcal{F}$ is reversible.*

Proof. The local rule of the inverse OCA τ can be defined by

- $\tau(z^{(1)}, y^{(k+1)}) = x^{(n)}$ for all $x, y \in X$ such that $\delta(x,y) = z$;
- $\tau(x^{(i)}, *) = x^{((i-1 \mod n)+1)}$ otherwise. ∎

The idea of the embedding is to desynchronize adjacent columns by shifting them vertically of some constant k between 1 and $n-1$. When two consecutive columns are not correctly arranged, they do not interact.

Lemma 6. *There exists an injective transformation of spacetime diagrams of $\mathcal{F}$ (in $X^{\mathbb{Z}\times\mathbb{N}}$) into spacetime diagrams of $\mathcal{F}'$ (in $X'^{\mathbb{Z}\times\mathbb{Z}}$): for every $c \in X^{\mathbb{Z}}$, there exists a unique configuration $c' \in X'^{\mathbb{Z}}$ for $\mathcal{F}'$ with*

$$\forall m \in \mathbb{Z}, p \in \mathbb{N}, \forall 1 \leq i \leq n, \mathcal{F}'^{m(n-k)+pn+i-1}(c')_m = (\mathcal{F}^p(c)_m)^{(i)}$$

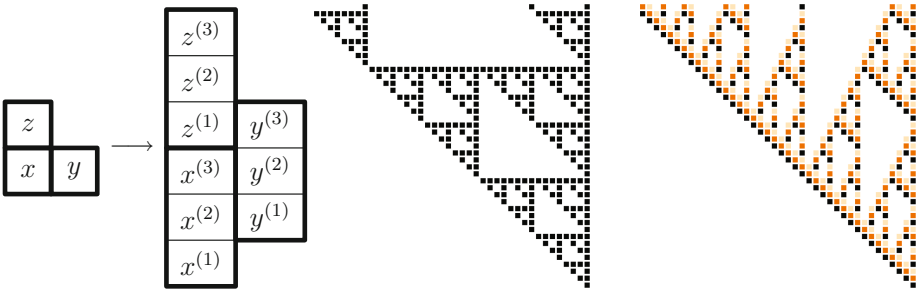


Fig. 2. The $(3, 1)$-embedding is given by the transformation of the local rule (to the left). The space-time diagram of the XOR OCA with a unitary configuration (at the center) is transformed into a space-time diagram of a reversible OCA (to the right).

Remark 1. If the OCA has a particular state 0 such that $\delta(x, 0) = x$ for every $x \in X$, we can keep a unique version of state 0 by identifying the $0^{(i)}$ as a unique state 0 where

$$\begin{aligned} \delta'(0, y^{(k)}) &= (\delta(0, y))^{(1)} \\ \delta'(0, *) &= 0 \text{ where } * \text{ can be anything} \\ \delta'(x^{(i)}, 0) &= x^{(1+(i \mod n))} \end{aligned}$$

This lemma allows to transfer results from $\mathcal{F}$ to $\mathcal{F}'$, in the sequel we prove the following result for permutive OCA and obtain it for reversible ones:

Reachability problem. *Given a reversible OCA with a quiescent state* $0 \in X$ *and two states* $x, y \in X$*, decide if* y *appears in the spacetime diagram generated by the initial configuration* ${}^{\omega}0.x0^{\omega}$.

Theorem 1. *The reachability problem is undecidable for reversible OCA.*

5 Main Construction

To prove the theorem, we need to embed some Turing complete computation into permutive OCA. This goal is achieved by simulating multi-head walking automata. This section describes the simulation of these automata by permutive OCA.

5.1 Multi-head Walking Automata

A multi-head walking automaton consists of a finite number of heads on the discrete line, each one of them provided with a state out of a finite set. At each step, they can only interact (read the state) with the heads that share the same cell, update their state and move. Initially, all the heads are in position 0.

Definition 2. *A* k-head walking automaton *is defined by* $(\Sigma, I, F, (f_i, g_i)_{1\leq i\leq k})$ *where* Σ *is a finite alphabet that does not contain* $\perp$, $I \in \Sigma$ *is the initial state,* $F \subseteq \Sigma$ *is a set of final states,* $\forall i, f_i : (\Sigma \cup \{\perp\})^k \to \Sigma$ *and* $\forall i, g_i : (\Sigma \cup \{\perp\})^k \to \{-1, 0, 1\}$ *are the update functions for the state and the position.*

A configuration of this automaton is a k*-tuple* $a = (a_i, s_i)_{1\leq i\leq k} \in (\mathbb{Z} \times \Sigma)^k$ *and its image configuration is* $(a_i + g_i(b_i), f_i(b_i))_{1\leq i\leq k}$ *where* $\forall 1 \leq i \leq k, b_i(j) = s_j$ *if* $a_i = a_j$ *and* $\perp$ *otherwise.*

Starting from configuration $(0, I)^k$*, the automaton computes the successive images and stops only if the position of every head is* 0 *and their states are final.*

A multi-head walking automaton can mimic a counter with 2 heads. Turing completeness is achieved by simulating 2-counter Minsky machines.

Proposition 2. *The halting problem of* 4*-head walking automata is undecidable.*

5.2 Finite Configurations

Every finite configuration can be written $c = {}^\omega 0.u_0 0^\omega$ for some $u_0 \in X^n, n \in \mathbb{N}$, hence its successive images can only be $\mathcal{F}^k(c) = {}^\omega 0 v_k.w_k 0^\omega$ for some $v_k \in X^k$, $w_k \in X^n$, that is: the finite non-quiescent word extends to the left, one cell at each step. We also have the following result.

Lemma 7 (Coven *et al.* [6]). *In the spacetime diagram associated to a finite configuration, every column is periodic.*

5.3 P-signals

Most cellular automata constructions use signals as elementary geometrical building blocks. Unfortunately, it is not possible to embed a classical signal in a permutive OCA nor is it possible to directly simulate multi-head automata. We provide P-signals as a technique to replace signals of speed k by an (n, k)-embedding of the front line of the spacetime diagram of the XOR with only one non-zero cell. This will effectively allow to send a signal through space in a reversible permutive OCA.

For our construction, we fix $n = 7$, hence we can use speeds from 1 to 6. Due to Remark 1, the states of these signals will be denoted $\mathbb{Z}_8 = \{0, 1, 2, 3, 4, 5, 6, 7\}$. Next lemma states that different speeds induce similar spacetime diagrams up to a vertical shift on each column.

Lemma 8. *Denote* $\mathcal{F}_k$ *and* $\mathcal{F}_{k'}$ *the OCA corresponding to signals of speeds* k *and* k' *and let* c *be the configuration* ${}^\omega 0.10^\omega$ *then we have*

$$\forall m \leq 0, \forall p \geq 0, \mathcal{F}_k^p(c)_m = \mathcal{F}_{k'}^{p-(k-k')m}(c)_m.$$

5.4 2-recognizability

Our construction relies on the ability to identify some specific sets of spacetime positions. We achieve this goal by considering products of independent P-signals. To avoid boring calculations, we rely on the following lemma to assert some regularity of P-signals and rely on the theory of p-recognizable sets of tuples of integers [5] to characterize these positions.

Lemma 9. *The spacetime sequence $\Delta : \mathbb{N}^2 \to \mathbb{Z}_{n+1}$ of the (n,k)-embedding $\mathcal{F}$ of the XOR, where $\Delta(x,y) = \mathcal{F}^y({}^\omega 0.10^\omega)_{-x}$, is 2-recognizable.*

Proof. The spacetime sequence Δ of the (n,k)-embedding $\mathcal{F}$ is generated by the 2-substitution $s : \mathbb{Z}_{n+1} \to \mathbb{Z}_{n+1}^2$ uniquely defined, as per Fig. 2, by

$$s(0) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \qquad s\begin{pmatrix} n \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} n \cdots (k+1) & k \cdots 1 & n \cdots (k+1) & \cdots & 1 \\ 0 \cdots \quad 0 & n \cdots \cdots \cdots & 1 & 0 \cdots 0 \end{pmatrix}^{\mathsf{T}}$$

Indeed, the substitution rule is compatible with the local rule of the OCA. ■

5.5 Computation Windows

Let $\mathcal{F}$ denote the OCA we are constructing. We use 4 P-signals of speeds 2, 3, 5 and 6 as foundations of $\mathcal{F}$ — they allow us to build computation windows. Let c be the initial configuration which is null everywhere except for $c(0) = (1,1,1,1)$. We use Lemma 9 twice for each of the following lemmas, that is, we have a characterization of the set of positions where some specific state pairs appear. Both lemmas are illustrated in Figs. 3, 4 and 5.

Lemma 10

$$\forall m \leq 0, p \geq 0, \begin{cases} \mathcal{F}^p(c)_m = (1,0,\_,\_) \\ \mathcal{F}^p(c)_{m+1} = (0,3,\_,\_) \end{cases} \Leftrightarrow \begin{cases} \exists h \in \mathbb{N}^*, m = -8^h \\ p \equiv -3m - 1[-14m] \end{cases}$$

Lemma 11

$$\forall m \leq 0, p \geq 0, \begin{cases} \mathcal{F}^p(c)_m = (\_,\_,7,6) \\ \mathcal{F}^p(c)_{m+1} = (\_,\_,5,0) \end{cases} \Leftrightarrow \begin{cases} \exists h \in \mathbb{N}^*, m = -8^h \\ p \equiv -12m - 1[-14m] \end{cases}$$

Hence we add a fifth layer using alphabet $\{0,1\}$ with the rule:

$$\begin{aligned} \delta_5\left((1,0,\_,\_,x),(0,3,\_,\_,\_)\right) &= 1 - x \\ \delta_5\left((\_,\_,7,6,x),(\_,\_,5,0,\_)\right) &= 1 - x \\ \delta_5\left((\_,\_,\_,\_,x),(\_,\_,\_,\_,\_)\right) &= x \quad \text{otherwise.} \end{aligned}$$

Actually, the same arguments work for columns $-2{\cdot}8^h, h \in \mathbb{N}^*$ and $-4{\cdot}8^h, h \in \mathbb{N}^*$, hence we use them all. We therefore call computation windows the vertical segments in the spacetime diagram where the fifth layer contains 1:

$$(m,p) \text{ in a computation window } \Leftrightarrow \begin{array}{l} \exists h \in \mathbb{N}^*, m = -2^h, p' \equiv p[14 \cdot 2^h] \\ \text{and } 3 \cdot 2^h \leq p' < 12 \cdot 2^h. \end{array}$$

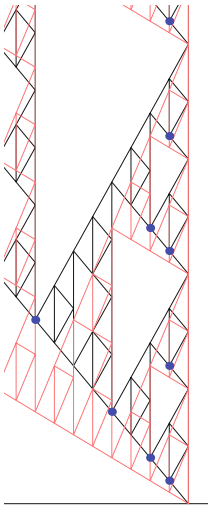


Fig. 3. Speeds 2 and 3 P-signals determine the positions of the lower points of the computation windows as stated in Lemma 10.

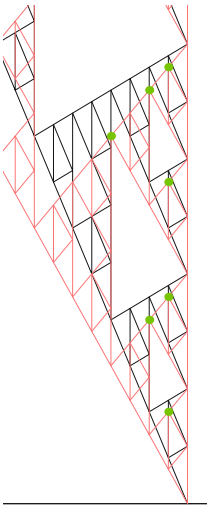


Fig. 4. Speeds 5 and 6 P-signals determine the positions of the upper points of the computation windows as stated in Lemma 11.

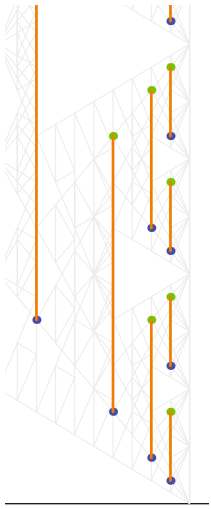


Fig. 5. In light gray, speeds 2, 3, 5 and 6 P-signals together allow to draw the whole computation windows.

5.6 Computing Heads

Every head of the walking automaton is simulated on a new layer. It is composed of a support that is mainly a speed 4 P-signal, and an internal state that belongs to the set of states Σ of the walking automaton. We need 4 heads, hence there are 4 additional layers (6 to 9).

The idea is that the heads move like speed 4 P-signal carrying internal states until they reach a computation window (as illustrated in Fig. 6). Then, depending on the context, they update their internal state and eventually shift upward or downward. Each shift corresponds to a move of the head of the walking automaton. Hence its position is encoded by the global shift applied to it, call it the *height* of the head.

More formally, each layer representing a head uses the alphabet $([1..7] \times \Sigma) \cup (0, \perp)$. Outside computation windows, the first component follows the rule of speed 4 signals, and the second component is maintained if possible or otherwise taken from the right neighbor.

Suppose now that we compute the new state of cell m at time p with (m, p) inside a computation window. Denote by x, y and z the states of cells (m, p), $(m+1, p)$ and $(m, p+1)$ in the spacetime diagram. The following rules apply to layer 6 (the same applies for other layers):

– if $y_6 = (0, \perp)$, then apply standard rules.

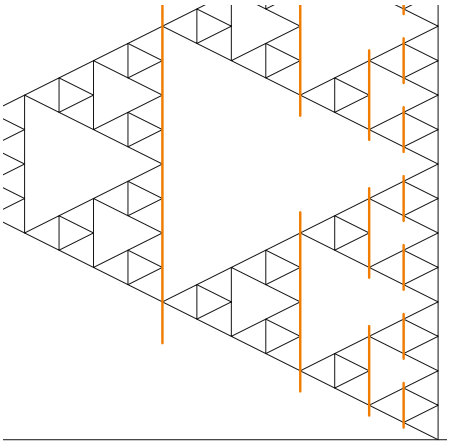


Fig. 6. The heads are supported by speed 4 P-signals that pass through the computation windows. The windows are large enough so that this property remains even after they are shifted up or down.

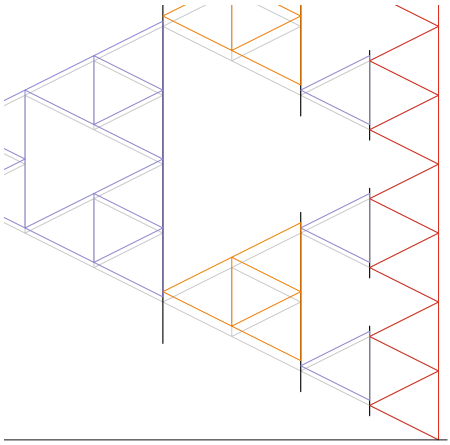


Fig. 7. When the head arrives in a computation window, it is here shifted upward, then upward again and downward in the third window. The state also changes then (red, later blue, orange and blue again). In light gray, another head whose position does not change. (Color figure online)

– if $y_6 \neq (0, \perp)$, look at every other layer where the support state is the same and apply f_1 and g_1, the update functions of the walking automaton, with these states. If the result of g_1 is -1 (resp. 0, 1), apply the rule of a speed 3 (resp. 4, 5) signal to the first component. If the support of z_6 is not 0, then the internal state is given by f_1.

5.7 Simulation

Finally, the OCA we build has 9 layers: 4 P-signals to determine the computation windows on the fifth layer, and 4 to simulate the 4 heads of the Turing universal walking automaton. The initial configuration is the finite configuration c with

$$\begin{aligned} c(0) &= (1, 1, 1, 1, 0, (1, I), (1, I), (1, I), (1, I)) \\ c(m) &= (0, 0, 0, 0, 0, (0, \perp), (0, \perp), (0, \perp), (0, \perp)) \quad \text{elsewhere.} \end{aligned}$$

First note that the height can vary of at most 1 each time the head crosses a column $-2^h, h \in \mathbb{N}$, hence:

Lemma 12. *Given any simulating head, while its height is between $-h$ and h in column $m = -2^h, h \in \mathbb{N}$, every non $(0, \perp)$ value of the layer in this column is inside a computation window.*

This means that each head has to apply the update rule when arriving in column $-2^h, h \in \mathbb{N}$. Now check that every time it does, one step of the computation of the walking automaton is simulated. The main point is to ensure that two heads of the walking automaton are on the same cell at step h if and only if the support of their simulating heads coincide on column $2^h - 1$. This is true since, for any head whose height is s, its support takes value 1 exactly in cells $(2^h - 1, p)$ with $p \equiv \big(4 \cdot (2^h - 1) + s\big)\,[7 \cdot 2^h]$. This property is due to the XOR rule, with the $(7, 4)$-embedding. Figure 7 illustrates this behaviour with two heads (one does not move to simplify the representation). This completes the simulation and the proof of Theorem 1.

6 One Step Further

Orbit periodicity problem. *Given a reversible OCA with a quiescent state $0 \in X$ and a state $x \in X$, decide if the spacetime diagram generated by the initial configuration $^\omega 0.x 0^\omega$ is periodic.*

Theorem 2. *The orbit periodicity problem is undecidable for reversible OCA.*

If the computation halts, the windows and the 4 P-signals that help determine the computation heads keep progressing eternally. To reach periodicity, it is necessary (and enough thanks to Lemma 7) to kill them all. It is possible to do so by giving killing orders to the heads. At the halting step h_0, each head is given the responsibility to kill one of the P-signals while sacrificing itself. They have to slow down or speed up to meet the corresponding P-signal. Again, we use Lemma 9 to prove that the meeting can happen on column 2^{h_0+1} with a local context that does not happen elsewhere.

References

1. Akhavi, A., Klimann, I., Lombardy, S., Mairesse, J., Picantin, M.: On the finiteness problem for automaton (semi) groups. Int. J. Algebra Comput. **22**(06), 1250052 (2012)
2. Amoroso, S., Patt, Y.N.: Decision procedures for surjectivity and injectivity of parallel maps for tessellation structures. J. Comput. Syst. Sci. **6**(5), 448–464 (1972)
3. Bartholdi, L., Silva, P.V.: Groups defined by automata. In: AutoMathA Handbook (to appear). https://arxiv.org/abs/1012.1531
4. Boyle, M., Maass, A., et al.: Expansive invertible onesided cellular automata. J. Math. Soc. Jpn. **52**(4), 725–740 (2000)
5. Bruyere, V., Hansel, G., Michaux, C., Villemaire, R.: Logic and p-recognizable sets of integers. Bull. Belg. Math. Soc. Simon Stevin **1**(2), 191–238 (1994)
6. Coven, E., Pivato, M., Yassawi, R.: Prevalence of odometers in cellular automata. Proc. Am. Math. Soc. **135**(3), 815–821 (2007)
7. Dartnell, P., Maass, A., Schwartz, F.: Combinatorial constructions associated to the dynamics of onesided cellular automata. Theoret. Comput. Sci. **304**(1–3), 485–497 (2003)

8. Delorme, M., Mazoyer, J., Ollinger, N., Theyssier, G.: Bulking II: classifications of cellular automata. Theoret. Comput. Sci. **412**(30), 3881–3905 (2011)
9. Gillibert, P.: The finiteness problem for automaton semigroups is undecidable. Int. J. Algebra Comput. **24**(01), 1–9 (2014)
10. Glushkov, V.M.: The abstract theory of automata. Uspekhi Matematicheskikh Nauk **16**(5), 3–62 (1961)
11. Hedlund, G.A.: Endomorphisms and automorphisms of the shift dynamical systems. Math. Syst. Theory **3**(4), 320–375 (1969)
12. Kari, J., Ollinger, N.: Periodicity and immortality in reversible computing. In: Ochmański, E., Tyszkiewicz, J. (eds.) MFCS 2008. LNCS, vol. 5162, pp. 419–430. Springer, Heidelberg (2008). doi:10.1007/978-3-540-85238-4_34
13. Kari, J.: The nilpotency problem of one-dimensional cellular automata. SIAM J. Comput. **21**(3), 571–586 (1992)
14. Kari, J.: Theory of cellular automata: a survey. Theoret. Comput. Sci. **334**(1), 3–33 (2005)
15. Klimann, I.: Automaton semigroups: the two-state case. Theory Comput. Syst. **58**(4), 664–680 (2016)
16. Šunić, Z., Ventura, E.: The conjugacy problem in automaton groups is not solvable. J. Algebra **364**, 148–154 (2012)