Journées Automates Cellulaires 2008 (Uzès), pp. 54-64

SOFIC ONE HEAD MACHINES

A. GAJARDO

Departamento de Ingeniería Matemática, Universidad de Concepción,
Casilla 160-C, Concepción, Chile
E-mail address: anahi@ing-mat.udec.cl
URL: http://www.ing-mat.udec.cl/~anahi

Abstract. There are several systems consisting in an object that moves on the plane by following a given rule. It is frequently observed that these systems eventually fall into an unexplained repetitive movement. The general framework of k-dimensional Turing machines with only one head is adopted. A subshift is associated to each Turing machine, and its properties are studied. The subshift consists in the set of sequences of symbols that the machine reads together with the states that it has through each evolution. The focus is placed on the machines whose associated subshift is sofic. These machines cannot make long tours, i.e., the time between two consecutive visits to a given cell is bounded, and this property characterises them. It is proved that all of these machines eventually fall into a repetitive movement when starting over an initially periodic coloration. Nevertheless, it seems that the machines with a sofic subshift are too simple. Many known machines remain out of scope. As an example, the 0,1 and 2 pebble automata with 1 symbol are studied.

Introduction

We call "One Head Machine" an automaton that lives in a discrete space. It can walk, read and write symbols, and its behaviour is governed, at discrete time, by a deterministic and finitely described rule. Examples of this kind of dynamical systems are the Langton's Ant [9, 8, 6], the Pebble Automata [2], the one head Turing Machines [3, 7] and the Lorentz Lattice Gas [1, 10]. Such an object can represent a particle that collides with obstacles; a living being that interacts with its environment; an automaton that performs a task, etc.

All of these systems are more or less comprised by the following definition.

Definition 0.1. A *One Head Machine* over $\mathbb{Z}^k$ is a 4-tuple (S, Q, k, δ) where:

- S is a finite set, representing the state of the environment at each lattice point, and called *symbol set*,
- Q is a finite set, representing the internal state of the machine, called *state set*,
- $k \in \mathbb{N}$ represents the dimension of the lattice,

Key words and phrases: Multidimensional Turing Machine, Formal Language, Symbolic System.
This work has been supported by CONICYT FONDECYT #1061036.

- $\delta = (\delta_S, \delta_Q, \delta_D)$ is the *transition function*, where $\delta_i : S \times Q \to i$, for each $i = S$, Q or D, and $D = \{\pm e_j\}_{j=1}^k$ are the k canonical vectors in $\mathbb{Z}^k$ together with their opposites.

The elements of $\mathbb{Z}^k$ are called *cells*. A configuration of the system is given by an assignment of symbols to each cell, $c : \mathbb{Z}^k \to S$, called coloration; a position $g \in \mathbb{Z}^k$; and a state $q \in Q$, i.e., the phase space is $X = S^{\mathbb{Z}^k} \times \mathbb{Z}^k \times Q$.

The global transition function $T : X \to X$ is defined by $T((c, g, q)) = (c', g', q')$, where

- $q' = \delta_Q(c(g), q)$,
- $g' = g + \delta_D(c(g), q)$,
- $c'(g) = \delta_S(c(g), q)$ and $c'(u) = c(u)$ for all $u \neq g$.

This system can be fruitfully studied by projecting it into a symbolic system [4, 5, 7], as we precise in the next definition. This method works well due to a relevant feature of this system: all the changes happen only on the machine position, the rest of the coloration remaining static. Thus, if we register the sequence of symbols that the machine reads together with its state, we describe the entire evolution of the system without ambiguity.

Definition 0.2. Given a one head machine $M = (S, Q, k, \delta)$ and its associated dynamical system (X, T), let $\pi : X \to S \times Q$ be defined by $\pi(c, g, q) = (c(g), q)$ and let $\psi : X \to (S \times Q)^{\mathbb{N}}$ be defined by $\psi(x) = (\pi(T^n(x)))_{n\in\mathbb{N}}$. The *t-shift* of (X, T) is $S_T = \psi(X)$.

The set $\psi(X)$ represents all the possible sequences of pairs (symbol, state) that the machine can produce when considering all the possible initial configurations. Given an infinite sequence $y = \binom{\alpha_1\alpha_2\cdots}{q_1q_2\cdots} \in S_T$, we can deduce the machine itinerary. In fact, if we suppose that the initial position is 0, its position at iteration j must be:

$$I(y)_j = \sum_{i=1}^{j-1} \delta_D(\alpha_i, q_i) \quad (\forall 2 \leq j) \quad , \quad I(y)_1 = 0$$

and the set of visited cells is given by

$$V(y) = \{I(y)_j | 1 \leq j\}$$

The initial symbol of the visited cells can be deduced from y and it is given by the following formula

$$c_y(g) = \alpha_i \quad \text{where} \quad i = \min\{j | I(y)_j = g\} \quad (\forall g \in V(y))$$

The partial function c_y is a kind of pre-image of y by ψ in the following sense: if c is an extension of c_y to $\mathbb{Z}^k$ then $\psi(c, 0, q_1) = y$. This means that the sequence $\psi(x)$ contains information about the visited cells and discards the symbols of the other cells. Moreover, it is invariant under translations.

Remark 0.3. $I(y)$, $V(y)$ and c_y can be defined also if y is a finite word. In this context the following properties hold for every u, $v \in (S \times Q)^*$:

(1) $I(uv)_j = I(u)_j$, if $j \leq |u| + 1$.
(2) $I(uv)_j = I(u)_{|u|+1} + I(v)_{j-|u|}$, if $j \geq |u| + 1$.

The set S_T is sensitive to many of the machine properties. For example, if $\psi(x)$ is periodic we can deduce that the sequence of movements of the machine is periodic, i.e., that the machine is making a regular movement (which can be propagative or cyclic).

Next section recall some concepts from symbolic dynamics and presents basic properties of S_T. In Section 2, we characterise the machines having a sofic t-shift and we prove an important feature of these systems: starting over a periodic coloration with a finite number of perturbations the machine always finishes by falling in a periodic movement. The last section shows how to adapt the this theory to a particular type of pebble automaton, we obtain two results already proved by Delorme and Mazoyer [2].

1. Basic notions and previous results

Given a finite set Σ, the set $\Sigma^{\mathbb{N}}$ denotes the set of infinite sequences of elements of Σ. A function σ is defined on $\Sigma^{\mathbb{N}}$ by: $\sigma(y_1y_2y_3...) = y_2y_3y_4...$, it is called the *shift* function. A metric can be defined on $\Sigma^{\mathbb{N}}$ by: $d(y,z) = 2^{-n}$, where n is the smallest index such that $y_n \neq z_n$. This metric makes $\Sigma^{\mathbb{N}}$ compact and σ continuous. Closed and σ invariant sets are called *subshifts*.

The finite sequences of elements of Σ are called *words*. The set of words is denoted by Σ^*. If a word $v \in \Sigma^*$ appears as a subsequence of an infinite sequence $y \in \Sigma^{\mathbb{N}}$, it is called a *factor* of y, and this is denoted by $v \sqsubseteq y$. A *language* is any subset of Σ^*.

Any subshift $Y \subset \Sigma^{\mathbb{N}}$ has an associated language $L(Y) \subset \Sigma^*$, defined by:

$$L(Y) = \{w \in \Sigma^* : (\exists y \in Y)\ w \sqsubseteq y\}.$$

This is the *factors language* of Y. The factors language characterises Y because $Y = \{y \in A^{\mathbb{N}} : (\forall u \sqsubseteq y)\ u \in L(Y)\}$.

Another way to characterise a subshift is through a set of forbidden words. A language P is a *set of forbidden words* for Y if

$$Y = \{y \in A^{\mathbb{N}} : (\forall u \sqsubseteq y)\ u \notin P\}.$$

If Y has a finite set of forbidden words, Y is said to be a *shift of finite type (SFT)*.

The complexity of Y is defined with reference to the complexity of its language $L(Y)$. For instance, Y is said to be *sofic* if $L(Y)$ is regular. It is easy to see that if Y is a SFT, Y is also sofic.

Let us come back to the shift associated to a one head machine: S_T. It is a subshift. Moreover, functions T, σ and ψ satisfy $\psi \circ T = \sigma \circ \psi$. The following result characterises the words in $L(S_T)$. This property can be easily proved by induction.

Lemma 1.1. [7] *If $w = \binom{\alpha_1..\alpha_n}{q_1..q_n} \in L(S_T)$, then for all $i \in \{1,..,n\}$:*

$$q_{i+1} = \delta_Q(\alpha_i, q_i) \qquad \textit{(state coherence)} \tag{1.1}$$

and for any pair $1 \leq i < j \leq n$, such that $I(w)_i = I(w)_j$ (say $= g$) and for every k between i and j, $I(w)_k \neq g$, one has that

$$\alpha_j = \delta_S(\alpha_i, q_i). \qquad \textit{(writing coherence)} \tag{1.2}$$

Moreover, these are sufficient conditions for w to belong to $L(S_T)$.

Equation (1.1) expresses that the sequence of states must be coherent with the transition rule of the machine. Equation (1.2) expresses that when the machine visits a cell a second time, it must find the symbol that it wrote there when it visited it the first time. A set of forbidden words for S_T can be obtained from these two equations. From Equation (1.1) we obtain the following set:

$$P_1 = \left\{ \begin{pmatrix} \alpha\beta \\ qp \end{pmatrix} : p \neq \delta_Q(\alpha, q) \right\}$$

Equation (1.2) refers to trajectories that visit two times the same cell.

Definition 1.2. A word $w \in L(S_T)$ whose itinerary starts and finishes in the same cell and does not visit that cell in between, is called a *cycle*, i.e., w is a cycle if it satisfies: $I(w)_1 = I(w)_{|w|+1} = 0$ and $I(w)_j \neq 0$, for all $j \in \{2, .., |w|\}$. When saying "the cycle w" we will being making reference to either the word w, the itinerary $I(w)$ or the set $V(w)$; the interpretation will be clear from the context.

There is a set of forbidden words for each cycle $w = \binom{\alpha_1..\alpha_n}{q_1..q_n}$: the word $\binom{\alpha_1..\alpha_n\beta}{q_1..q_n q}$ is forbidden for every $\beta \neq \delta_S(\alpha_1, q_1)$. Thus, we obtain the following set of forbidden words:

$$P_2 = \left\{ \begin{pmatrix} \alpha_1..\alpha_n\beta \\ q_1..q_n q \end{pmatrix} : \begin{pmatrix} \alpha_1..\alpha_n \\ q_1..q_n \end{pmatrix} \text{ is a cycle and } \beta \neq \delta_S(\alpha_1, q_1) \right\}.$$

S_T is defined by the set of forbidden words $P = P_1 \cup P_2$. P_1 is finite but P_2 may be infinite, depending on the behaviour of the machine. For example, if the machine never visits a cell more than once, P_2 is empty. If the number of cycles is finite, P_2 is finite. In both cases, the set of forbidden words of S_T is finite and therefore S_T is a SFT. With some work it is possible to prove the converse [7], i.e., if S_T is a SFT, then the number of cycles is finite. If S_T is a SFT, the machine has significant movement restrictions that prevent it from making long cycles. What happens when S_T is sofic? In [7] the one dimensional case was studied and it was established that if S_T is sofic, then it is of finite type too. Is this also true in Z^k? What is the relation between the complexity of S_T and the machine behaviour? In order to answer these questions we need more information about the relation between S_T, T and the automaton that recognises S_T. Let us recall some definitions.

Definition 1.3. A Deterministic Finite Automaton (DFA) is a 5-tuple $M = (A, \Omega, \lambda, o_0, F)$ where A is the *input alphabet,* Ω *is the* states set, $\lambda : A \times \Omega \to \Omega$ is a partial function called *transition function*, $o_0 \in \Omega$ is the *initial state and* F *is the set of* final states.

A labelled graph, G_M, is associated to M. Its set of vertices is Ω, and the label of an edge (e, f) is 'a' if and only if $f = \lambda(a, e)$.

The language recognised by M consists of all words w in A^* such that there exists a path in G_M with label w, starting on vertex o_0 and finishing on a vertex $f \in F$.

If a language L is the factors language of some subshift, then it is closed for the factor relation, i.e., if $w \in L$ and u is a subword of w, then $u \in L$. Consequently, the automaton $M = (S \times Q, \Omega, \lambda, o_0, F)$ that recognises it can be chosen such that $F = \Omega$. In the following we will omit the set F from the automata definition.

Definition 1.4. A language is said to be *regular if it is recognised by some DFA.*

Remark 1.5. Given a vertex $\nu \in \Omega$, it holds that:

(1) If ν has an input edge labelled by (α, q), then all the exiting edges of ν have a label of the form $(\beta, \delta_Q(\alpha, q))$, with $\beta \in S$, because the next state of the machine is uniquely determined by α and q and it is $\delta_Q(\alpha, q)$.

(2) Since every word in $L(S_T)$ defines a unique path in G_M, Equation (1.2) implies that ν has only one exiting edge if and only if every path from o_0 to ν corresponds to an itinerary that has already visited its last cell. Otherwise, ν has exactly $|S|$ exiting edges.

(3) The last assertion is not valid when $|S| = 1$. But, in this case, S_T is of finite type, more precisely, it is a finite set composed by eventually periodic sequences. The cycles exist in a finite quantity. The vertices of the automaton that recognises it have degree 1 (except for o_0)

This remark allows us to characterise the one head machines whose t-shift is sofic.

2. When S_T is sofic

Let us consider a metric in $\mathbb{Z}^k$ as follows. Given two points: $p = (p_1, p_2, .., p_k)$ and $g = (g_1, g_2, .., g_k)$, the distance between p and g is

$$d(p, g) = \sum_{i=1}^{k} |p_i - g_i|.$$

Thus the set $B(p, n) = \{g \in \mathbb{Z}^k : d(p, g) \leq n\}$ represents the ball of radius n and centre p.

Lemma 2.1. *The number of cycles of $L(S_T)$ is finite if and only if the distance that a cycle can attain form the origin is bounded.*

Proof. Let us suppose that every cycle is contained in a ball of radius n. The number of configurations defined in this ball is $|S|^{2n^k} \times 2n^k \times |Q|$. Each configuration corresponds to at most one cycle, hence there is a finite number of cycles. ■

The following theorem has already been proved for $k = 1$ in [7].

Theorem 2.2. *S_T is sofic if and only if the number of cycles of $L(S_T)$ is finite.*

Proof. In one direction, the result is trivial since every SFT is sofic.

Let us suppose that S_T is sofic. Therefore, there exists a DFA $M = (S \times Q, \Omega, \lambda, o_0)$ that recognises it and satisfies the conditions given in Remark 1.5. Let us suppose that the length of the cycles is arbitrary large. Lemma 2.1 implies that for every natural n there is a cycle that attains a distance bigger than n from its initial cell: 0. Let us consider the set of cycles that makes this for $n = |\Omega|$. Let us choose from this set the shortest cycle $w = w_1..w_m = \binom{\alpha_1..\alpha_m}{q_1..q_m}$.

Given $r < m$, we can use Remark 0.3, with $u = w_1..w_{r-1}$ and $v = w_r..w_m$ and the fact that $I(w)_{|w|+1} = 0$ to obtain that:

$$I(w)_r = -I(w_r..w_m)_{m-r+1} \neq 0. \tag{2.1}$$

The cycle w corresponds to a unique path in the graph G_M: $o_0 o_1, .., o_m$. From Remark 1.5, the vertex o_m has an exit degree equal to 1, because the last cell of w (cell 0) has already been visited.

Let $l \leq m$ be such that $d(I(w)_l, 0) > n$. Since for every j, $d(I(w)_j, 0) \leq j - 1$, we can assert that $l-1 > n$. In consequence, there must exist two repeated vertices between o_1 and o_{l-1}: $o_i = o_j$. Thus $o_0 o_1..o_i o_{j+1}..o_m$ is also a path in G_M, its label is $u = w_1 w_2..w_i w_{j+1}..w_m$, its length is $t = m - j + i$, and $I(u)_{t+1}$ has already been visited. This implies that there exists r such that $I(u)_r = I(u)_{t+1}$. By using Remark 0.3 over u decomposed by $u_1..r_{r-1}$ and $u_r..u_t$, we obtain that $I(u_r..u_t)_{t-r+2} = 0$, i.e., $u_r..u_t$ is a cycle of length $t - r + 1$.

If $r > i$, $u_r..u_t = w_{r+j-i}..w_m$ which, from Equation (2.1), is not a cycle, therefore $r \leq i$. In summary, $1 \leq r \leq i < j < l < m$. Now, we use Remark 0.3 again over $u_r..u_{k-1}$ and $u_k..u_t$, with $k = l - j + i$, and we obtain

$$\begin{array}{rcll} I(u_r..u_{k-1})_{k-r+1} & = & -I(u_k..u_t)_{t-k+1} & \\ & = & -I(w_l..w_m)_{m-l+1}, & \text{since } k > i, \\ & = & I(w)_l, & \text{due to Equation (2.1).} \end{array}$$

Hence $d(I(u_r..u_t)_{k-r+1}, 0) = d(I(w)_l, 0) > n$. We conclude that $u_r..u_t$ is a cycle that attains a distance bigger than n and is shorter than w, which is a contradiction. ■

This implies that the machines whose t-shift is sofic are very simple: they cannot revisit far cells; the diversity of cycles they can do is finite; its t-shift is also a SFT; and all the closed itineraries can be putted inside a finite ball. Since the state of the visited cells can be interpreted as the external "remembers" of the machine, this ball represents its attainable memory. Sofic machines have in fact a finite memory.

Corollary 2.3. *If S_T is sofic and $V(\psi(c, g, q))$ is infinite, then the number of times that the machine visits each cell is bounded by a finite constant which only depends on the machine.*

Proof. Let r be the radius of a ball containing all the closed trajectories. The number of configurations defined on this ball is finite, say N. Let $x = (c, g, q)$ be an initial configuration. If some cell p is visited more than N times during the evolution of T on x, we can assert that the machine remained inside the ball of radius r and centre p from the first to the last time that it visited p. Within this time, some configuration of the ball has appeared two or more times. Which means that the system has fallen in a periodic point and that its complete itinerary is contained in a finite set. ■

When a sofic machine starts over a periodic coloration, its behaviour is particularly simple, as the following theorem establishes. This theorem is proved in [7] for $k = 1$.

Theorem 2.4. *If S_T is sofic and c is periodic except for a finite number of cells, then $\psi(c, g, q)$ is eventually periodic for every $g \in \mathbb{Z}^k$ and $q \in Q$.*

Proof. We can suppose, without loss of generality, that the initial position of the machine is 0. Let q_0 be its initial internal state, and $c : \mathbb{Z}^k \to S$ a periodic coloration except for a finite number of cells. This means that c is equal to some periodic coloration d except for a finite set of cells E.

Now, let us study $y = \binom{\alpha_i}{q_i}_{i \in \mathbb{N}} = \psi(c, 0, q_0)$. Two possibilities appear: $V(y)$ can be finite or not. If it is finite, $(T^i(c, 0, q_0))_{i \in \mathbb{N}}$ is eventually periodic, and so is y.

Let us analyse the case when $V(y)$ is infinite. Let n be the last iteration in which a cell of E is visited. This means that after iteration n every cell g either has been already visited or its state is given by $d(g)$.

Coloration d consists in the repetition of a pattern defined on a rectangle R. Let us assume that $R = \{0, .., r_1\} \times \{0, .., r_2\} \times \cdots \times \{0, .., r_k\}$, where $r_i \in \mathbb{N}$ for all i. This means that the value of d at cell $g = (g_1, g_2, .., g_k)$ is equal to $d(g_1 \text{mod } r_1, g_2 \text{mod } r_2, .., g_k \text{mod } r_k)$.

Now let $M = (S \times Q, \Omega, \lambda : (S \times Q) \times \Omega \to \Omega, o_0)$ be the automaton that recognises S_T. We define $\overline{M} = (S \times Q, \Omega \times R, \overline{\lambda}, (o_0, 0))$ where $\overline{\lambda} : (S \times Q) \times (\Omega \times R) \to (\Omega \times R)$ is defined by:

$\overline{\lambda}(\binom{\alpha}{q}, \binom{\mu}{f}) = \binom{\nu}{g} \Leftrightarrow \lambda(\binom{\alpha}{q}, \mu) = \nu$ and $(\forall i)\ f_i = (g_i + \delta_D(\alpha, q)_i) \bmod r_i$.

$\overline{M}$ recognises the same language than M. Moreover, it registers the position of the machine modulo R, i.e., if $((o_i, g^i))_{i\in\mathbb{N}}$ is the sequence of vertices in $G_{\overline{M}}$ whose label is y, then, for every $j \in \mathbb{N}$, $I(y)_{j+1} = g^j$ modulo R.

From Remark 1.5, we distinguish two kinds of vertices in $G_{\overline{M}}$: either $deg((o_i, g^i)) = 1$ or $deg((o_i, g^i)) = |S|$. If $deg((o_i, g^i)) = |S|$, we know that $I(y)_{i+1}$ is being visited by the first time at iteration $i+1$. If in addition $i > n$, we can assert that $c(I(y)_{i+1}) = d(I(y)_{i+1}) = d(g^i)$. Consequently, the label of the arc $((o_i, g^i), (o_{i+1}, g^{i+1}))$ is $d(g^i)$, this means that (o_{i+1}, g^{i+1}) is uniquely determined by (o_i, g^i). Thus starting from (o_n, g^n) only one sequence of vertices of $G_{\overline{M}}$ can be taken, hence $((o_i, g^i))_{i\in\mathbb{N}}$ is ultimately periodic and so is y. ∎

These two theorems can be easily proved for other regular grids than $\mathbb{Z}^k$, for example, Cayley graphs of groups.

3. Pebble automata with one symbol

Pebble automata are two dimensional one head machines that cannot write but are provided with a set of "pebbles" that they can drop and recover in order to mark their way. They could be seen as a particular kind of one head machine by assimilating the pebbles as part of the internal state and space symbols. The following definition is adapted from [2] for the particular case where the space has only one symbol.

Definition 3.1. A *Pebble Automata* is a 3-tuple (Q, δ, l) where:

- Q is a finite set, representing the internal state of the machine,
- l represents the number of pebbles, and
- $\delta = (\delta_P, \delta_Q, \delta_D)$ is the transition function, where
  - $\delta_P : Q \times \{0,1\}^l \times \{0,1\}^l \to \{0,1\}^l$ determines the pebbles that will be taken or dropped,
  - $\delta_Q : Q \times \{0,1\}^l \times \{0,1\}^l \to Q$ determines the new machine state, and
  - $\delta_D : Q \times \{0,1\}^l \times \{0,1\}^l \to D$ determines the moving direction of the machine.

Moreover, δ_P satisfies that for every $q \in Q$ and $p, r \in \{0,1\}^l$, $\delta_P(q, p, r) \leq p + r$; this assures that the machine can only act over pebbles that the machine is carrying or that are on the current machine position.

The configuration of the system is given by an assignment of pebbles to each cell $c : \mathbb{Z}^2 \to \{0,1\}^l$, a position $g \in \mathbb{Z}^2$, a state $q \in Q$, and the pebbles reserve of the machine $p \in \{0,1\}^l$. Thus, the phase space is $X = (\{0,1\}^l)^{\mathbb{Z}^2} \times \mathbb{Z}^2 \times Q \times \{0,1\}^l$.

The global transition function $A : X \to X$ is defined by $A((c, g, q, p)) = (c', g', q', p')$, where

- $q' = \delta_Q(q, c(g), p)$,
- $p' = (p + \delta_P(q, c(g), p)) \bmod 2$,
- $g' = g + \delta_D(q, c(g), p)$,
- $c'(g) = (c(g) + \delta_P(q, c(g), p)) \bmod 2$ and for all $u \neq g$, $c'(u) = c(u)$.[1]

The system starts with the empty configuration $c(u) = (0, 0, .., 0)$ and all the pebbles on the machine: $p = (1, 1, .., 1)$.

[1]let us remark that $c(g) + p = c'(g) + p'$.

In [2], the behaviour of these automata is studied for the case in which the number of symbols is 1. The object of doing this is to analyse the ability of the automaton to explore the plane without external help. They prove that this task is impossible when the automaton has less than 3 pebbles. Delorme and Mazoyer prove this by showing that the d-pebble automata have serious restrictions on their movements when $d \leq 2$. This can be illustrated within the present theory too. 0-pebble automata are a trivial case. Since they have no pebble, they are actually one head machines where $|S| = 1$, and $k = 2$. Thus the number of cycles is bounded and, independently from the initial state $q_0 \in Q$, the machine will always finish by making repetitive movements.

3.1. 1-pebble automata

As we said before, pebble automata can be seen as a particular case of one head machine by assimilating the pebbles as part of the internal state of the machine and symbols of the space. This means to define a one head machine $M = (\overline{S}, \overline{Q}, 2, \overline{\delta})$ where $\overline{Q} = Q \times \{0,1\}$, $\overline{S} = \{0,1\}$ and $\overline{\delta}$ is defined appropriately. The second component of the state represents the pebble reserve of the machine. The symbol represents the pebble content of the cells.

The difference between pebble automata and one head machines is that in the pebble automata the total number of pebbles in the system is fixed and constant. A 1-pebble automaton is a one head machine that works over a particular set of configurations: those with exactly one pebble in the whole space.

Thus the shift of a pebble automata is smaller than the shift of its corresponding one-head machine. In the pebble automata, not only cycles define forbidden words. If the pebble automata put the pebble somewhere, it cannot find it elsewhere. This fact induces additional forbidden words. First, P_0 forbids to find the pebble in the plane when the pebble is on the machine.

$$P_0 = \left\{ \begin{pmatrix} \alpha \\ \overline{q} \end{pmatrix} : \overline{q} = (q, 1) \wedge \alpha = 1 \right\}.$$

Second, let $w = \binom{\alpha_1..\alpha_n}{\overline{q}_1..\overline{q}_n} \in (\{0,1\} \times \overline{Q})^*$ be such that $I(w)_i \neq 0$ for every $i \geq 2$. In this case, we know that if at the beginning of w the initial cell contains the pebble, i.e., $\alpha_1 = 1$, and the machine does not take it ($\delta_P(\alpha_1, q_1) = 0$), then $\alpha_i = 0$ for all $i \geq 2$. The same happens if the machine has the pebble and drops it at the beginning, i.e., if $\alpha_1 = 0 \wedge \delta_P(\alpha_1, q_1) = 1$. In these cases, we can add the forbidden word $\binom{\alpha_1..\alpha_n 1}{\overline{q}_1..\overline{q}_n \overline{q}_{n+1}}$. This defines the following set of forbidden words:

$$P_3' = \left\{ w = \begin{pmatrix} \alpha_1..\alpha_n 1 \\ \overline{q}_1..\overline{q}_n \overline{q}_{n+1} \end{pmatrix} : \alpha_1 + \delta_P(\alpha_1, q_1) = 1 \wedge (\forall i \geq 2)\ I(w)_i \neq 0 \right\}.$$

We only need to consider a finite part of P_3', because if the machine has no pebble, it behaves like a 0-pebble automaton where the length of cycles is bounded, say by M, and therefore longer trajectories do not visit 0. Thus forbidden words longer than M can be replaced by the word $\binom{00..01}{\overline{q}_1..\overline{q}_M}$. We obtain a new set of forbidden words:

$$P_3 = \left\{ w = \begin{pmatrix} \alpha_1..\alpha_n 1 \\ \overline{q}_1..\overline{q}_n\overline{q}_{n+1} \end{pmatrix} : \alpha_1 + \delta_P(\alpha_1, q_1) = 1 \wedge (\forall i \geq 2)\ I(w)_i \neq 0 \wedge n \leq M \right\}$$
$$\bigcup \left\{ \begin{pmatrix} 00..01 \\ \overline{q}_1..\overline{q}_M \end{pmatrix} : \overline{q}_1 = (q_1, 0) \right\}.$$

Finally, each time we have a word $w = \binom{0..01}{\overline{q}_1..\overline{q}_n}$ where $\overline{q}_1 = (q_1, 0)$, we know that this pebble must have been dropped by the automaton at some past iteration, then w is a suffix of a cycle that begins by leaving the pebble on the plane. We thus obtain the forbidden set P'_4.

$$P'_4 = \left\{ \begin{pmatrix} 0..01 \\ \overline{q}_1..\overline{q}_n \end{pmatrix} : \overline{q}_1 = (q_1, 0) \wedge (\forall v \in C_0) \begin{pmatrix} 0..0 \\ \overline{q}_1..\overline{q}_{n-1} \end{pmatrix} \not\preccurlyeq v \right\}.$$

Where C_0 denotes the set of cycles that begins by leaving the pebble and $\preccurlyeq$ is the suffix relation. Again, only a finite part of P'_4 is enough, because we know that every word longer than M is not suffix of some cycle. Hence, we consider the set P_4:

$$P_4 = \left\{ \begin{pmatrix} 0..01 \\ \overline{q}_1..\overline{q}_n \end{pmatrix} : \overline{q}_1 = (q_1, 0) \wedge (\forall v \in C_0) \begin{pmatrix} 0..0 \\ \overline{q}_1..\overline{q}_{n-1} \end{pmatrix} \not\preccurlyeq v \wedge n \leq M \right\}$$

By avoiding these words, we assure that the pebble is always found exactly where it was dropped.

Let us recall now the definition of the set P_2:

$$P_2 = \left\{ \begin{pmatrix} \alpha_1..\alpha_n\beta \\ \overline{q}_1..\overline{q}_n\overline{q} \end{pmatrix} : \begin{pmatrix} \alpha_1..\alpha_n \\ \overline{q}_1..\overline{q}_n \end{pmatrix} \text{ is a cycle and } \beta \neq \alpha_1 + \delta_P(\alpha_1, \overline{q}_1) \bmod 2 \right\}$$

We distinguish three types of cycles, depending on the position of the pebble at the beginning of the itinerary: a) the pebble is left at 0 ($\alpha_1 + \delta_P(\alpha_1, \overline{q}_1) = 1$), b) the pebble is not at 0 nor on the machine ($\alpha_1 = 0$ and $\overline{q}_1 = (q, 0)$, $q \in Q$), c) the pebble is carried by the machine ($(\alpha_1 + \delta_P(\alpha_1, \overline{q}_1) = 0 \wedge \overline{q}_1 = (q, 1)) \vee \alpha_1 + \delta_P(\alpha_1, \overline{q}_1) = 2$).

There is only a finite number of cycles of type a). If the cycle is of type b), there are two cases. First, the pebble does not appear in the cycle, it behaves like a 0-pebble automaton, therefore there is only a finite quantity of these cycles. Second, the pebble is found during the cycle. In this case, by avoiding the forbidden words of P_3 and P_0 we prevent from finding the pebble at position 0. Finally, if the cycle is of type c), two cases appear again. First, the pebble is over the machine during the whole cycle; in this case, the set P_0 assures that the pebble will not be found at the initial cell. Second, the machine drops the pebble somewhere; in this case, by avoiding the words of P_3 we preclude the possibility of finding the pebble at 0. It follows that, only a finite number of words of P_2 are necessary. The union of P_0, P_1, P_3, P_4 and a finite part of P_2 is a forbidden set for S_A. We conclude that S_A is of finite type.

Remark 1.5 is not valid in this case because we are using information that is not available for general one head machines. But Remark 1.5 is used in Theorem 2.4 to have vertices with exit degree equal to one. In $L(S_A)$ only short words that do not contain the pebble can be enlarged in two different ways, because words longer than M that has no the pebble will never have it; and if a word contain the pebble, its future is uniquely determined. This means that after iteration M, all the attainable vertices has degree 1. Thus the proof of Theorem 2.4 applies to the present case and we can conclude that the automaton will always finish by making a repetitive movement.

3.2. 2-pebble automata

In these automata, trajectories as those of Figure 1 can occur, as it is shown in [2]. Theorem 2.2 can be used to assert that, in this case, S_A is not sofic. We can wonder about the complexity of S_A. Delorme-Mazoyer proved that these automata cannot explore the plane, this suggest that its complexity cannot be very high.

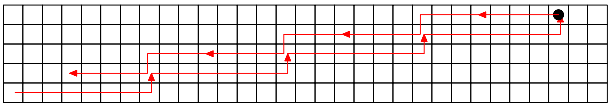

Figure 1: A trajectory that may contain cycles of arbitrary length. The automaton comes from below, it finds the pebble on top and comes back.

4. Discussion

The results of this paper apply to very simple machines. Sometimes, proving the hypothesis of our theorems is not easy. Nevertheless, we think that the present paper is a first step in the construction of effective tools to understand the dynamics of one head machines.

The study of pebble automaton was not easy, the principal reason is that pebble automaton do not feet with the definition of one head machine. Delorme and Mazoyer obtained the same results through an analysis of similar complexity

The next step may be to study machines with a context free language. In such a machine, the cycles may be very long, but they probably have a simple structure. Maybe the 2-pebble automata with one symbol is in this class. It would be important to study the dynamical properties of these machines.

Langton's Ant is a one head machine that, apparently, always falls in a periodic movement when it starts with a coloration with a finite number of black cells. Unhappily the cycles that Langton's Ant can do can be very complicated and arbitrarily long, hence our results do not apply. But maybe the ideas presented in the proofs may help to understand Langton's Ant behaviour.

Acknowledgement

The author wishes to acknowledge the referees for their careful reading and their deep comments.

References

[1] L. A. Bunimovich and S. Troubetzkoy. Topological dynamics of flipping Lorentz lattice gas models. *J. of Stat. Physics*, 72:297–307, 1993.

[2] M. Delorme and J. Mazoyer. Pebble automata. figures families recognition and universality. Technical Report 32, Ecole Normale Supérieure de Lyon, Lyon,France, 1999.

[3] A. K. Dewdney. Computer recreations: Two-dimensional Turing machines and tur-mites make tracks on a plane. *Scientific American*, pages 124–127, September 1989.

[4] A. Gajardo. *Dependence of the behavior of the dynamical system Langton's ant on the network topology.* PhD thesis, Universidad de Chile and École normale supérieure de Lyon, 2001.

[5] A. Gajardo. A symbolic projection of Langton's ant (extended abstract). In *Proc. (Discrete Models for Complex Systems) (DM-CS'03)*, volume AB, pages 57–68, 2003.

[6] A. Gajardo and E. Goles. Dynamics of a class of ants on a one-dimensional lattice. *Theor. Comput. Sci.*, 322(2):267–283, 2004.

[7] A. Gajardo and J. Mazoyer. One head machines from a symbolic approach. *Theor. Comput. Sci.*, 370:34–47, 2007.

[8] A. Gajardo, A. Moreira, and E. Goles. Complexity of Langton's ant. *Discrete Applied Mathematics*, 117:41–50, 2002.

[9] C. G. Langton. Studying artificial life with cellular automata. *Physica D*, 22:120–149, 1986.

[10] A. Quas. Infinite paths in a Lorentz lattice gas model. *Probab. Theory Rel*, 114(2):229–244, 1999.