Available online at www.sciencedirect.com
ScienceDirect
Discrete Applied Mathematics 155 (2007) 2130–2139
DISCRETE APPLIED MATHEMATICS
www.elsevier.com/locate/dam

A single-copy minimal-time simulation of a torus of automata by a ring of automata

Bruno Martin*, Claudine Peyrat

I3S, Université de Nice–Sophia-Antipolis, CNRS, 2000 route des lucioles, B.P. 121, F-06903 Sophia-Antipolis Cedex, France

Received 21 September 2004; received in revised form 4 September 2006; accepted 14 May 2007
Available online 25 May 2007

Abstract

We consider cellular automata on Cayley graphs and we simulate the behavior of a torus of $n \times m$ automata (nodes) by a ring of $n \cdot m$ automata (cells). Our simulation technique requires the neighborhood of the nodes to be preserved. We achieve this constraint by copying the contents of nodes on the cells. We consider the problem of minimizing the number of the copies. We prove that it is possible to simulate the behavior of a torus on a ring with a single copy on each cell if and only if n and m satisfy a given condition. In that case we propose a time-optimal algorithm. We thus improve a previous work done by Martin where two copies were requested. When the condition on n and m is not fulfilled one can use the previous algorithm.

Keywords: Cellular automata; Cayley graphs; Complexity

1. Introduction

A cellular automaton (CA for short) is a network of identical finite state machines which work in parallel and synchronously. The interconnection network is also required to be regular and can even be considered as a Cayley graph. Finite state machines are placed on the vertices of the graph, and they communicate with their nearest neighbors through the edges. Some papers [9,16,18,1,7] have already considered this generalization of cellular automata.

Our goal is to simulate the behavior of an automaton T with $n \times m$ *nodes* arranged on a $n \times m$ torus (with its five neighbors, including the node itself) by an automaton R with $n \cdot m$ *cells* arranged on a ring (with only three neighbors, including the cell itself). We will describe our simulation method in detail later. However our simulation technique requires the preservation of the neighborhoods of T. As each node of T has four neighbors while a cell of R only two, we will consider a *multi-layer* ring automaton for which each cell c is formed of $2 + k$ *layers* composed as follows: a layer where the simulation takes place, a synchronization layer (for technical purposes) and k layers containing the copies of other nodes of T. In order to be able to apply our simulation technique we require that if cell c represents the node v, the state of all neighbors of v in T is copied in some layer of either c or one of the two neighbors of c in R.

With the above conditions, we want to minimize k (the number of copies) and the simulation time.

Clearly our conditions may not be fulfilled if k equals 0. The main goal of this paper is to study the case $k = 1$ and prove that such a simulation exists if and only if $\gcd(n, m) = 2$. Depending upon the case, the time-overhead of our

* Corresponding author. Tel.: +33 492 965 107; fax: +33 492 965 155.
E-mail address: Bruno.Martin@unice.fr (B. Martin).

0166-218X/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2007.05.019

simulation is lower-bounded by $\min\{n+m, nm/2\}$ times that on the torus, and is the best possible for that kind of simulation. The case $k=2$ was already considered in [9]; it also shows that the above restriction on n and m can be abolished and a torus automaton with $n \times m$ nodes can be simulated by a ring of $n \cdot m$ cells with a time-overhead $O(\min\{m, n\})$ times that on the torus. Also observe that the latter complexity bound is not a lower bound. The other question we address here is whether the simulation of a torus of automata by a ring of automata can be realized with less resources in terms of number of states. We answer positively but this new simulation imposes restrictions on n and m. When these restrictions are fulfilled, a lower bound on the simulation time is given. It thus provides another example of a lower-bound result for cellular automata, which are not very common.

The paper is organized as follows. Section 2 introduces some definitions and notations on cellular automata and Cayley graphs, the commonly agreed definition of a simulation, the definition of multi-layer cellular automata, a notion introduced by Martin in [8] which generalizes partitioned cellular automata introduced by Morita in [14] and some classical synchronization results which help to rapidly design some algorithms. Section 3 presents our neighbor-preserving alternating Hamiltonian cycle decomposition of a torus, which is the keypoint of the paper. We also state that there is a unique way to embed a torus into a ring under single copy assumption. We then compute the distance between a node and its copy (the notion will be detailed later) and design a simple simulation of a torus automaton by a ring of automata which strongly depends on a synchronization result.

2. Definitions and notation

2.1. Cellular automata and Cayley graphs

Cellular automata on Cayley graphs have been introduced to prove a generalization of the Moore and Myhill theorem [1,7].

Let $\Gamma = \langle \mathscr{G} | \mathscr{R} \rangle$ be a finitely presented group with $\mathscr{G}$ the set of *generators* not containing the identity 1_Γ and $\mathscr{R}$ the set of *relators* [2]. For every group presentation Γ, there is an associated *Cayley (di)graph* $G = (\Gamma, E)$; the graph vertices are the elements of Γ and the arcs are colored by the generators in the following way. There exists an arc colored with generator g from a vertex x to a vertex y if and only if $y = xg$ in Γ. A *path* in G is a sequence of vertices, each adjacent to the next or, equivalently, a starting vertex together with a word over $\mathscr{G}$. A *cycle* is a path containing at least three vertices such that the last vertex is adjacent to the first or, equivalently a starting vertex together with a word over $\mathscr{G}$ which reduces to 1_Γ. A *Hamiltonian cycle* is a cycle which contains all the vertices of G exactly once. Given x and y two vertices of a Cayley graph G, the *distance* between x and y is the length of a minimal path from x to y and is denoted by $d_G(x, y)$.

A CA over a Cayley digraph G is a 4-tuple $\mathscr{A} = (Q, G, N, \delta)$ for which we associate a cell of the CA to each vertex of the digraph. Q denotes the finite set of the states, $G = (\Gamma, E)$ is the Cayley digraph of a finitely presented group Γ, N the neighborhood is such that $N = \mathscr{G} \cup \mathscr{G}^{-1} \cup \{1_\Gamma\}$ and $\delta : Q^{|N|} \to Q$ is the local transition function which updates the state of cell i at time t according to the states of its neighbors at time $t-1$. We also define a distinguished state denoted by q, the *quiescent state* which has the property to remain quiescent by δ: $\delta(q, \ldots, q) = q$. Note that we only consider the von Neumann neighborhood defined as the set of vertices at distance at most one from every cell (thus including the cell itself). We define a *configuration* of the CA as an application c which attributes a state to each cell. The set of all the configurations of a CA is denoted by $\mathbb{C} = Q^\Gamma$ on which the *global function* Δ of the CA is defined by means of the local transition function δ, $\forall c \in \mathbb{C}$, $\forall x \in \Gamma$,

$$\Delta(c)(x) = \delta(c(xg_{|\mathscr{G}|}^{-1}), \ldots, c(xg_1^{-1}), c(x), c(xg_1), \ldots, c(xg_{|\mathscr{G}|}),$$

As in [16,17], a CA on a $m \times n$ torus is defined over the Cayley graph of the group presentation $\Gamma_T^{n,m} = \{\alpha, \beta \mid \alpha\beta = \beta\alpha, \alpha^n = 1_\Gamma, \beta^m = 1_\Gamma\}$; it has $n \times m$ nodes and will be denoted by $T_{m,n}$. A CA on a l-ring is defined over the Cayley graph of the group presentation $\Gamma_R^l = \{w \mid w^l = 1_\Gamma\}$; it has l cells and will be denoted by R_l. Other kinds of cellular automata on Cayley graphs can be defined as in [6,17] like, for instance, on hexagons or triangles. For more examples of Cayley graphs which can serve to define cellular automata, one can refer to [3].

One of the purposes of this paper is to find out a decomposition of the torus into a Hamiltonian cycle and use this cycle to define a ring. We will see later that only a special kind of Hamiltonian cycle is useful. We call it *alternating Hamiltonian cycle*; it is a Hamiltonian cycle whose word over the set of generators is labeled $(\alpha\beta)^\star(\alpha+\varepsilon)$ or equivalently

$(\beta\alpha)^\star(\beta+\varepsilon)$, $(\alpha^{-1}\beta^{-1})^\star(\alpha^{-1}+\varepsilon)$, $(\beta^{-1}\alpha^{-1})^\star(\beta^{-1}+\varepsilon)$ where ε denotes the empty word. Then, we will use this ring of automata to simulate the behavior of the torus of automata.

2.2. Simulation

Below, we recall a commonly used step by step simulation between two cellular automata, as defined in [4] and used in [17]. It expresses that if the ring of automata R simulates each step of the behavior of the torus of automata T in τ units of time, there must exist a correspondence between the corresponding configurations of R and T:

Definition 1. Let $\mathbb{C}_R$ and $\mathbb{C}_T$ be, respectively, the sets of configurations of CA R and T. We say that R simulates each step of T in time τ if there exist two recursive functions $\kappa : \mathbb{C}_T \to \mathbb{C}_R$ and $\rho : \mathbb{C}_R \to \mathbb{C}_T$ such that $\kappa \circ \rho = \mathrm{Id}$ and for all $c, c' \in \mathbb{C}_T$, there exists $c'' \in \mathbb{C}_R$ such that if $\Delta_T(c) = c'$, $\Delta_R^\tau(\kappa(c)) = c''$ then $\rho(c'') = c'$, where Δ_M denotes the global transition of CA M and Δ_M^t the tth iterate of the global transition of CA M.

Observe that Definition 1 is more suitable when considering computation problems rather than decision problems for which a simpler notion of simulation can be considered.

The main idea of the paper is to construct a function κ which preserves the homogeneity of the neighborhood.

The design of the simulation of the torus of automata by a ring of automata will be made easier with *multi-layer* CA which have been introduced in [8] and strongly uses a minimal-time solution to the Firing Squad Synchronization Problem, which is recalled in Section 2.4.

2.3. Multi-layer cellular automata

Definition 2. A *ℓ-layer CA* is a CA $\mathscr{C} = (\ell, Q, G, N, \delta)$ where

- ℓ denotes the number of layers;
- Q is the Cartesian product of ℓ sets of states $Q = \prod_{i=0}^{\ell-1} Q_i$. In the sequel, we denote an element of Q by $\underline{x} = (x_0, x_1, \ldots, x_{\ell-1})$;
- G, a Cayley graph of the group presentation Γ;
- $N = \mathscr{G} \cup \mathscr{G}^{-1} \cup \{1_\Gamma\}$. Let $n = |N|$;
- $\delta : Q^n \to Q$ such that $\delta(\underline{x^0}, \underline{x^1}, \ldots, \underline{x^{n-1}}) = \gamma(\delta_0(x_0^0, x_0^1, \ldots, x_0^{n-1}), \delta_1(x_1^0, x_1^1, \ldots, x_1^{n-1}), \ldots, \delta_{\ell-1}(x_{\ell-1}^0, x_{\ell-1}^1, \ldots, x_{\ell-1}^{n-1}))$ for $x_i^j \in Q_i (0 \leqslant i < \ell, 0 \leqslant j < n$ and $\underline{x} = (x_0, \ldots, x_{\ell-1}))$. The function γ is used to define a global interaction between the layers since the δ_i's only consider the corresponding layer of the neighborhood.

In the sequel we want to simulate $T_{m,n}$ by R_l, a multi-layer CA. Our goal in the paper is to minimize the number of layers as well as the simulation time.

2.4. Firing Squad Synchronization Problem

The Firing Squad Synchronization Problem (FSSP) is due to Myhill (1957) and can be expressed as: "Given a line of n soldiers, how can they fire simultaneously knowing that the firing order, coming from one general at one end of the line, needs a certain time to propagate". FSSP has a solution in time $3n-1$ by Minsky [13] and was proved time-optimal for time $2n-1$ by Mazoyer in [11]. If two generals are allowed, the synchronization time decreases to n [12]. It also has generalizations as, for instance, to Cayley graphs [18].

Lemma 1 (*Firing Squad Lemma*). *There exists a CA $G_1 = (1, Q, G, N, \delta)$ with G the graph presentation of $\mathbb{Z} = \{w\}$ and $N = \{w^{-1}, 1_\mathbb{Z}, w\}$. Its cells are arranged on a non-quiescent line of n cells with special symbols $g, F \in Q$ and a quiescent state q such that: $\Delta^t({}^\omega q g 0^{n-1} q^\omega) = {}^\omega q F^n q^\omega$ for $t = 2n-2$ and $\Delta^t({}^\omega q g 0^{n-1} q^\omega)(i) \neq F$ for $0 \leqslant t < 2n-2$. Δ denotes the global transition function corresponding to the local transition function δ, ${}^\omega q$ the infinite repetition of state q to the left and q^ω the infinite repetition of state q to the right.*

State g represents the general. At the end, all the non-quiescent cells are in state F.

Remark 1. Given a solution to the FSSP in time T, one can easily design a solution to the FSSP in time $aT + b$, given $a \in \mathbb{N}\backslash\{0\}$ and $b \in \mathbb{N}$.

The solution to the FSSP will be used in the following way. Once the torus of $n \times m$ automata has been "embedded" by κ on the ring of automata, the simulation can be done. Just after simulating one transition step of the torus, we run a global synchronization on the synchronization layer while a copy of the new state of every cell moves to its corresponding position on a auxiliary layer at the maximal speed of one cell by unit of time and stops as soon as the line of soldiers fires. This also gives the simulation time since the minimal distance between a node and its copy corresponds precisely to the synchronization time.

3. Single copy simulation

Theorem 1. *$T_{m,n}$ can be simulated by R_{mn} if and only if both m and n are such that* $\gcd(m, n) = 2$; *furthermore, when* $n \equiv 2 \bmod m$, *the number of layers of R_{mn} is minimal* (*a* $(1 + 2)$*-layer ring automata*) *and the simulation is done in time* $\Theta(n + m)$.

Theorem 1 is proved in several steps:

- If such a ring exists then there is an alternating Hamiltonian cycle in the torus automaton (Section 3.1).
- Find the convenient values of m and n to guarantee the existence of an alternating Hamiltonian cycle (Section 3.2).
- Arrange the copies of the nodes on the ring of nm cells (Section 3.3). Actually, only a single arrangement is possible, and therefore only one simulation can be defined.
- Find the minimal distance between a cell representing node v and the copy of v on the ring (Section 3.4). This distance will bound the time required to simulate the behavior of the torus of automata by the ring of automata.
- Sketch the behavior of the ring of nm cells (Section 3.5).

We introduce some notation. Let $T_{m,n} = (1, Q_T, G_T^{n,m}, N_T, \delta_T)$ be a CA on a torus of $n \times m$ nodes with $G_T^{n,m}$ the Cayley graph of the group presentation $\Gamma_T^{n,m}$ and $N_T = \{1_\Gamma, \alpha, \alpha^{-1}, \beta, \beta^{-1}\}$. Let $R_{mn} = (3, Q_R, G_R^{nm}, N_R, \delta_R)$ be a CA on a ring of nm cells with G_R^{nm} the Cayley graph of the group presentation Γ_R^{nm}, $N_R = \{1_\Gamma, w, w^{-1}\}$ and $\delta_R : (Q_R)^3 \to Q_R$. Q_R has two layers: $Q_R = (Q_T)^2 \times Q_{\text{Sync}} = (M, C, \text{Sync})$ where:

- M is the *main layer* which simulates a node v of $T_{m,n}$;
- C contains copies of other nodes of $T_{m,n}$;
- Sync is a *synchronization layer*, used for sending synchronously informations with the use of a solution to the FSSP.

Let $N_T(v)$ denote the neighbors of node $v = \alpha^i \beta^j$ on the torus according to Definition 2, and let $N_T^+(v) = N_T(v)\backslash\{v\}$ denote the set of neighbors of v not including v itself. Let c_v be the cell of the ring representing node v on the torus. Let $C(r)$ denote the copy-layer state-component of cell r on the ring.

3.1. Necessity of an alternating Hamiltonian cycle

Proposition 1. *If $T_{m,n}$ can be simulated by R_{mn} with a minimal number of layers* (*a* $(1+2)$*-layer ring automata*) *then there exists an alternating Hamiltonian cycle in $G_T^{n,m}$, the graph of the group $\Gamma_T^{n,m}$.*

Proof. Let us fix some node v, and let V_1 and V_3 be the neighbors of c_v on the ring. From our neighborhood constraint we get

$$N_T^+(v) = \{v\alpha, v\alpha^{-1}, v\beta, v\beta^{-1}\} \subset \{C(c_v), V_1, C(V_1), V_3, C(V_3)\} \tag{1}$$

Fact 1.

$$\{C(c_v), C(V_3)\} \cap N_T^+(v) \neq \emptyset,$$
$$\{C(c_v), C(V_3)\} \cap N_T^+(V_3) \neq \emptyset.$$

Fact 2. *V_1 and V_3 belong to $N_T^+(v)$.*

Proof. Suppose that $V_1 \notin N_T^+(v)$, then Eq. (1) becomes $\{v\alpha, v\alpha^{-1}, v\beta, v\beta^{-1}\} = \{C(c_v), C(V_1), V_3, C(V_3)\}$. Wlog, we may assume that $C(c_v) = v\alpha$. As $v \notin N_T^+(V_1)$, from Fact 1 applied to V_1 instead of c_v, we have $\{C(V_1), C(c_v)\} \subset N_T^+(V_1)$.

Suppose $C(V_1) = v\alpha^{-1}$. Then $\{v\alpha^{-1}, v\alpha\} \subset N_T^+(V_1)$, therefore $V_1 = v$, a contradiction. Therefore, $C(V_1) = v\beta$ or $C(V_1) = v\beta^{-1}$. Wlog, we assume $C(V_1) = v\beta$. From $\{C(V_1), C(c_v)\} \subset N_T^+(V_1)$, $C(c_v) = v\alpha$, we get $V_1 = v$ or $V_1 = v\alpha\beta$. Therefore $V_1 = v\alpha\beta$.

Consequently, we have $\{V_3, C(V_3)\} = \{v\alpha^{-1}, v\beta^{-1}\}$. Therefore, $\{C(c_v), C(V_3)\} \cap N_T^+(V_3) = \emptyset$, a contradiction to Fact 1. We conclude that $V_1 \in N_T^+(v)$.

Similarly, V_3 belongs to $N_T^+(v)$ as well. Wlog, we assume $V_1 = v\alpha^{-1}$. □

Fact 3. *If $V_1 = v\alpha^{-1}$, then $V_3 \neq v\alpha$.*

Proof. Suppose for contradiction that $V_3 = v\alpha$ then $\{v\beta, v\beta^{-1}\} \subset \{C(V_1), C(V_3), C(c_v)\}$. Wlog, we may assume that $v\beta \in \{C(V_1), C(V_3)\}$ and that $C(V_1) = v\beta$.

As $C(V_1) = v\beta \notin N_T^+(V_1)$, we deduce from Fact 1 that $C(c_v) \in N_T^+(V_1)$. Consequently, $v\beta^{-1} \neq C(c_v)$ and $v\beta^{-1}$ must be equal to $C(V_3)$.

Similarly, as $C(V_3) = v\beta^{-1} \notin N_T^+(V_3) = N_T^+(v\alpha)$, it follows from Fact 1 that $C(c_v) \in N_T^+(V_3)$.

We then have $C(c_v) \in N_T^+(V_1) \cap N_T^+(V_3)$, therefore $C(c_v) = v$ and V_3 does not satisfy the neighborhood constraint, a contradiction. □

Without loss of generality, we may assume that $V_3 = v\beta$. Consider V_5, the other neighbor of V_3 on the ring.

Fact 4. *If $V_1 = v\alpha^{-1}$, $V_3 = v\beta$, then $V_5 = v\alpha\beta$.*

Proof. From Fact 2 and 3 applied to V_3 instead of V, we must have $V_5 = v\alpha\beta$ or $V_5 = v\alpha^{-1}\beta$.

Suppose that $V_5 = v\alpha^{-1}\beta$ and $C(c_v) = v\beta^2$ (as $v\beta^2 \notin N_T^+(v)$ nor $v\beta^2 \notin N_T^+(V_1)$). Then $C(V_1)$ must belong to both $N_T^+(v)$ and $N_T^+(V_1)$, but $N_T^+(v) \cap N_T^+(V_1) = \emptyset$; a contradiction. Therefore, $C(c_v) \neq v\beta^2$. Similarly, $C(c_v) \neq v\alpha\beta$.

As a consequence, $\{C(V_3), C(V_5)\} = \{v\beta^2, v\alpha\beta\}$. But then $\{C(V_3), C(V_5)\} \cap N_T^+(V_5) = \emptyset$ contradicts Fact 1. □

We have proved that if we assume $V_1 = v\alpha^{-1}$, then $V_3 = v\beta$ and $V_5 = v\alpha\beta$.

Let us number the vertices of the torus $\{v_i\}_{1 \leqslant i \leqslant nm}$ according to their position on the ring. If we denote by v_{i+1} the neighbor of v_i different from v_{i-1} on the ring, we inductively have

$$v_{2i+1} = v\alpha^i\beta^i \quad \text{and} \quad v_{2i} = v\alpha^{i-1}\beta^i.$$

Therefore, $G_T^{n,m}$ contains an alternating Hamiltonian cycle, which completes the proof of Proposition 1. □

3.2. Convenient values of m and n

Proposition 2. *An alternating Hamiltonian cycle exists if and only if* $\gcd(n, m) = 2$.

Proposition 2 is proved with the next three facts:

Fact 5. *If* $\gcd(n, m) \neq 1$, *then an alternating Hamiltonian cycle exists if and only if* $\gcd(n, m) = 2$.

Proof. Let $\gcd(n, m) = d \geqslant 2$. If we describe the alternating Hamiltonian cycle, we have $\alpha^{(nm/d)}\beta^{(nm/d)} = 1_\Gamma$. It implies $(2nm/d) = nm$ and $d = 2$. Conversely, if $d = 2$, there exists an alternating Hamiltonian cycle, as $\alpha^{(nm/2)}\beta^{(nm/2)} = 1_\Gamma$, $\alpha^i\beta^i$ and $\alpha^{i+1}\beta^i \neq 1_\Gamma$ if $i < nm/2$. □

Fact 6. *If* $\gcd(n, m) = 1$ *with nm even*, *then an alternating Hamiltonian cycle does not exist.*

Proof. Suppose that $\gcd(n, m) = 1$ with nm even and that an alternating Hamiltonian cycle exists then, $\alpha^{(nm/2)}\beta^{(nm/2)} = 1_\Gamma$: therefore $\alpha^{(nm/2)} = 1_\Gamma$ and $\beta^{(nm/2)} = 1_\Gamma$. It follows that both n and m are even, a contradiction. □

Fact 7. *If* $\gcd(n, m) = 1$ *with nm odd, then an alternating Hamiltonian cycle does not exist.*

Proof. We first prove by induction on $n + m$ that we are in one of
Case 1: $\exists i : i + 1 \equiv 0 \bmod n$ and $i \equiv 0 \bmod m$ and $2i + 1 < nm$.
Case 2: $\exists i : i + 1 \equiv 0 \bmod m$ and $i \equiv 0 \bmod n$ and $2i + 1 < nm$.
For $n = 3$ and $m = 5$ (basis), we have $\alpha^6\beta^5 = 1_\Gamma$ and $2i + 1 < 15$. We assume it is true for $n + m$ relatively small and let (n, m) be such that wlog $n > m$. Then, by the Euclidean division algorithm, there exists p and r two integers such that $n = pm + r$ with $0 \leqslant r < m$. Since $\gcd(n, m) = 1$, $\gcd(m, r) = 1$ and, by inductive hypothesis, (m, r) is either in case 1 or in case 2.

- (m, r) is in case 1: $\exists i : i + 1 = \lambda m$ and $i = \mu r$ and $2i + 1 < mr$. Then, $2\mu r + 1 < mr$ and $\mu < (mr - 1)/2r < m/2$ and $2\lambda m - 1 < mr$ thus $\lambda < (mr + 1)/2m$. Let $i' = i + \mu pm = \mu r + \mu pm = \mu n$, $i' + 1 = \lambda m + \mu pm = m(\lambda + \mu p)$. We have $\alpha^{i'}\beta^{i'+1} = 1_\Gamma$. Then, $2i' + 1 = \mu(r + pm) + m(\lambda + \mu p) = \lambda m + \mu r + 2\mu pm < mr + mpm = m(r + pm) = mn$.
- (m, r) is in case 2: $\exists i : i + 1 = \mu r$ and $i = \lambda m$ and $2i + 1 < mr$. Let $i' = i + \mu pm = \lambda m + \mu pm$, $i' + 1 = \mu r + \mu pm = \mu n$. We have $\alpha^{i'}\beta^{i'+1} = 1_\Gamma$. Then, $2i' + 1 = \mu(r + pm) + m(\lambda + \mu p) < mn$.

Then, if we are either in case 1 or in case 2, an alternating Hamiltonian cycle does not exist. Indeed, assume we are in case 1; on a cycle starting with generator β, we return to β with a word of length $2i + 1$ since $\beta^{i+1}\alpha^{i+1} = \beta$ and on a cycle starting with generator α, we return to 1 with a word of length $2i + 1$ since $\alpha^{i+1}\beta^i = 1$. Case 2 is analogous. □

This completes the proof of proposition 2. □

3.3. Unicity

Since there exists an alternating Hamiltonian cycle for convenient values of m and n, how can we arrange the copies of the nodes to reconstruct the torus neighborhood on the ring? Next proposition gives the convenient arrangement of the cells:

Proposition 3. *If there exists an alternating Hamiltonian cycle, then* $T_{m,n}$ *can be simulated by* R_{nm} *with the following arrangement of cells*:

$$\xrightarrow{\beta} \begin{pmatrix} c_v(\alpha^{-1}) \\ \downarrow \alpha\beta^{-1} \\ c_v(\beta^{-1}) \end{pmatrix} \xrightarrow{\alpha} \begin{pmatrix} c_v \\ \downarrow \alpha^{-1}\beta \\ c_v(\alpha^{-1}\beta) \end{pmatrix} \xrightarrow{\beta} \begin{pmatrix} c_v\beta \\ \downarrow \alpha\beta^{-1} \\ c_v(\alpha) \end{pmatrix} \xrightarrow{\alpha} \begin{pmatrix} c_v(\beta\alpha) \\ \downarrow \alpha^{-1}\beta \\ c_v(\beta^2) \end{pmatrix} \xrightarrow{\beta}$$

with $c_v = \alpha^i\beta^j$ *the cell which represents node* (i, j).

Fact 8. *If* $C(c_v) \in N_T^+(v)$, *some neighbors on the torus are neither neighbors nor on the copy layer of neighbors on the ring.*

Proof. According to Fig. 2 the arrangement of the neighbors on the torus is as depicted in (a) or in (b) depending upon the position in the alternating Hamiltonian cycle. We only consider neighbors V_1, V_2 and V_3 of node v. If $C(c_v) \in N_T^+(v)$, then neither V_1 nor V_3 are neighbors of V_2 on the ring. Then, the cell representing V_2 lacks of information to achieve the transition. □

The arrangement of the copies is thus as described by Fig. 1. According to Fact 8, since V_2 is not a neighbor of V_1 and V_4 is not a neighbor of V_3, it implies $C(c_v)$ to be a neighbor common with V_1 and V_3 as shown on Fig. 1. One can assume that the arrangement of the copies on the ring must be of the form given by Proposition 3.

3.4. Determining the distance

Remains to find the distance between a node and its copy: let c_v be the cell representing the node v (i.e. $M(c_v) = c(v)$) and $c_\mu \neq c_v$ the cell such that $C(c_\mu) = c(v)$; we want to determine $d_G(c_v, c_\mu)$.

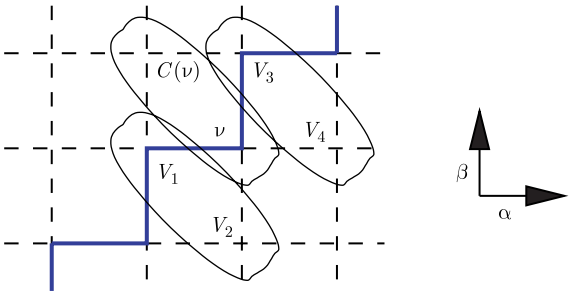

Fig. 1. Arrangement of the cells.

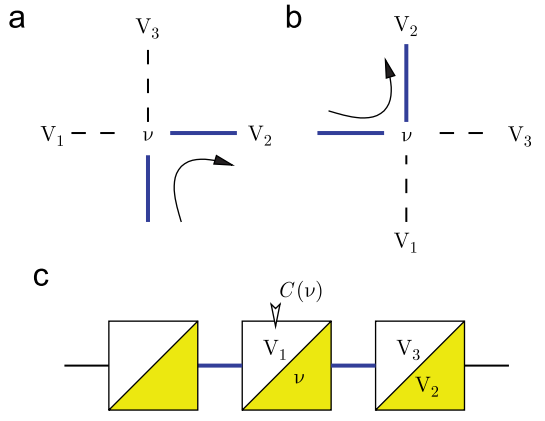

Fig. 2. Bad arrangement of the copies.

Proposition 4. *When both* m *and* n $(n \geqslant m)$ *are such that* $\gcd(n, m) = 2$, *the distance between a node and its copy on the alternating Hamiltonian cycle is lower-bounded by* $\min\{m + n, m \cdot n/2\}$ *and upper-bounded by* $\max\{m + n, m \cdot n/2\}$. *The distance is minimal when* $n \equiv 2 \bmod m$ *and equals* $2n - 2$.

Proof. Wlog let us assume $n \geqslant m$. We want to find the distance between a node and its copy on the ring. That is the distance between $\alpha^i \beta^j$ and $\alpha^i \beta^j \alpha^{-1} \beta$ or $\alpha^i \beta^j \alpha \beta^{-1}$. Writing the conditions on the alternating Hamiltonian cycle, we obtain for a cell at the end of α (i.e. labelled by a word of the form $((\alpha\beta)^\star + (\beta\alpha)^\star\beta)\alpha$) $\alpha^i \beta^j \beta^d \alpha^d = \alpha^i \beta^j \alpha^{-1} \beta$. For a cell at the end of β (labelled by a word of the form $((\alpha\beta)^\star + (\beta\alpha)^\star\beta)\alpha$) we get $\alpha^i \beta^j \alpha^d \beta^d = \alpha^i \beta^j \alpha \beta^{-1}$. Observe that the distance $2d$ between a node and its copy does not depend upon i nor j and that it must be even since m and n are even. Therefore, the Cayley graph with mn nodes is bipartite. We get the following relations for a cell at the end of α, i.e. labelled by a word of the form $((\alpha\beta)^\star + (\beta\alpha)^\star\beta)\alpha$ (Fig. 3):

$$\begin{cases} i + d \equiv i - 1 \bmod n \\ j + d \equiv j + 1 \bmod m \end{cases} \Leftrightarrow \begin{cases} d + 1 = \lambda n, \\ d - 1 = \mu m \end{cases}$$

and, at the end of β, i.e. labelled by a word of the form $((\alpha\beta)^\star\alpha + (\beta\alpha)^\star)\beta$:

$$\begin{cases} i + d \equiv i + 1 \bmod n \\ j + d \equiv j - 1 \bmod m \end{cases} \Leftrightarrow \begin{cases} d - 1 = \lambda n, \\ d + 1 = \mu m. \end{cases}$$

For a cell at the end of α, we have $2 = \lambda n - \mu m$. Since an alternating Hamiltonian cycle exists if and only if $\gcd(n, m) = 2$, there exists a, b such that $an - bm = 2$. The distance $2d = 2an - 2$ is minimal when $a = 1$, that is when $n \equiv 2 \bmod m$. Thus, the minimal distance between a node and its copy in the case of a alternating Hamiltonian cycle is $2n - 2$.

The proof for a cell at the end of β is analogous. □

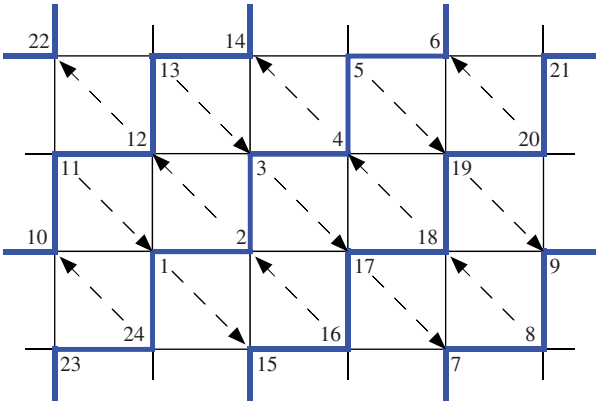

Fig. 3. Arranging the nodes of $T_{6,4}$.

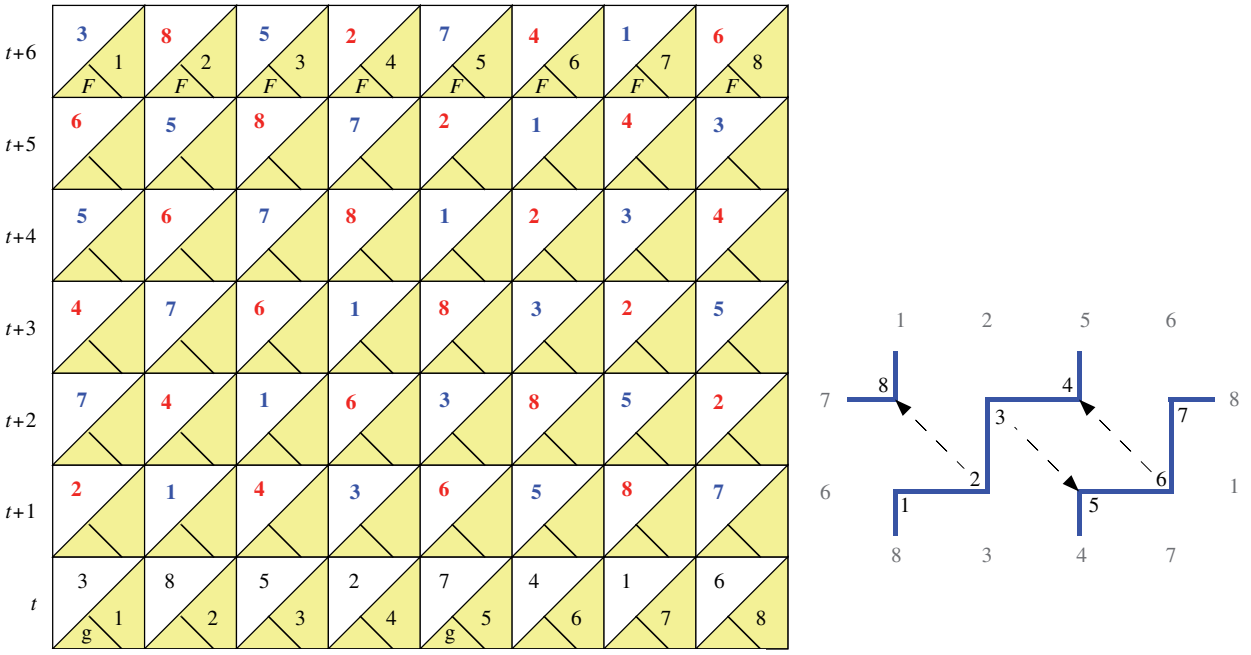

Fig. 4. Evolution of a ring of 4.2 cells.

Observe that in Fig. 4, the distance between a node and its copy is 6, which equals $m + n$ and reaches the minimum value, leading to a simulation in $2 \cdot n - 2$.

3.5. *Behavior of the ring*

The synchronization layer is as follows: there is a general on cells $1, n + 1, \ldots, (m - 1)n + 1$. The local transition function which rules this layer is obtained from a solution to the one-general FSSP (Lemma 1) in time $2an - 1$.

The simulation is inspired from [9] and is as follows:

(a) the contents of the main layer of odd cells are moved rightwards on layer C as long as no firing state appears on the synchronization layer,
(b) the contents of the main layer of even cells are moved leftwards on layer C as long as no firing state appears on the synchronization layer,

(c) when the firing states appear, the moved information is where it has to be and on the main layers R_{mn} can simulate one step of $T_{m,n}$. Afterwards, the moving process (a) and (b) is again performed.

The way to send information synchronously is detailed in [8].

Observe that the simulation is also time-optimal. This comes from the arrangement of the nodes of the torus on the ring of automata. In fact, the torus is separated into "slices" of n cells, the first one of each becoming a local general for the FSSP (or even of the generalized FSSP described in Remark 1) and the distance between a node and its copy is taken modulo m.

3.6. To synchronize or not to synchronize?

To synchronize or not to synchronize is a rather general question. A common belief while designing CA algorithms states that everything that can be done with synchronization can be done without as well. But the keypoint of this paper is the decomposition of the torus of automata into a neighbor-preserving alternating Hamiltonian cycle. And since the distance between a node and its copy agrees with the single general FSSP solution designed by Mazoyer, the easiest way to simulate the behavior of a torus of automata was to use synchronization. Nevertheless, with the use of *signals* (see [5]) and by moving the copies in the opposite direction, one can avoid synchronization. The synchronization-free algorithm would cut the ring into two halves, form a concurrent zig-zag between the boundaries which sets-up the emission of the copies. This algorithm would be more difficult to explain and requires more time than the one we propose in the present paper.

4. Conclusion

This paper improves a previous result by Martin [9]. We have minimized the number of copy requested to simulate the behavior of a torus of $n \times m$ automata by a ring of $n \cdot m$ automata. Observe that this number of copies cannot be further improved. It is thus the minimal number of copies requested to complete this task. Observe that the result forbids some values of n and m. This is the weakness of the present simulation. However, for these forbidden values, one can use the simulation proposed in [9] which works for any values.

In addition, this simulation together with a result from Róka [16], permits to prove in a straightforward way that cellular automata defined over a hexagonal grid can be simulated by cellular automata over a ring [10].

These results allow to discuss the relative power of cellular automata according to their interconnection network and let us state that torus of automata are strictly more powerful than rings of automata.

As an application, this simulation can serve to implement on parallel computers with a ring as underlying topology the computation of a 5-point Laplacian, an operation useful in image processing for edge-finding or for computing morphology-based operations. This computation is often approximated to save time.

Some questions remain open. If a synchronization-free simulation seems possible, what is its time-complexity? Furthermore, what is the complexity of the different solutions to this problem when considering the number of states or even the number of transitions, a resource complexity which is more and more taken into account (see for instance the recent paper by Noguchi [15]). Also, what is the consequence if we change the notion of simulation, relaxing the strong condition of effectively coding and decoding both configurations?

References

[1] T. Ceccherini-Silberstein, A. Machi, F. Scarabotti, Amenable groups and cellular automata, Ann. Inst. Fourier 49 (2) (1999) 673–685.
[2] P. Cohn, Algebra, Wiley, New York, 1989.
[3] H. Coxeter, W. Moser, Generators and Relations for Discrete Groups, Springer, Berlin, 1972.
[4] P. Dehornoy, Complexité et décidabilité, Springer, Berlin, 1993.
[5] M. Delorme, An introduction to cellular automata, Research Report 98-37, LIP ENS-Lyon (July 1998).
[6] M. Garzon, Models of massive parallelism: Analysis of cellular automata and neural networks, EATCS Monographs, Springer, Berlin, 1995.
[7] A. Machí, F. Mignosi, Garden of eden configurations for cellular automata on Cayley graphs of groups, SIAM J. Discrete Math. 6 (1) (1993) 44–56.
[8] B. Martin, A universal cellular automaton in quasi-linear time and its S-m-n form, Theoret. Comput. Sci. 123 (1994) 199–237.
[9] B. Martin, Embedding torus automata into a ring of automata, Internat. J. Found. Comput. Sci. 8 (4) (1997) 425–431.
[10] B. Martin, A simulation of cellular automata on hexagons by cellular automata on rings, Theoret. Comput. Sci. 265 (2001) 231–234.

[11] J. Mazoyer, A six-states minimal time solution to the firing squad synchronization problem, Theoret. Comput. Sci. 50 (1987) 183–238.
[12] J. Mazoyer, N. Reimen, A linear speed-up theorem for cellular automata, Theoret. Comput. Sci. 101 (1992) 59–98.
[13] M. Minsky, Finite and Infinite Machines, Prentice-Hall, Englewood Cliffs, 1967.
[14] K. Morita, Reversible simulation of one-dimensional irreversible cellular automaton, Theoret. Comput. Sci. 148 (1995) 157–163.
[15] K. Noguchi, Simple 8-state minimal time solution to the firing squad synchronization problem, Theoret. Comput. Sci. 314 (2004) 303–334.
[16] Z. Róka, One-way cellular automata on Cayley graphs, in: Proceedings of FCT'93, Lecture Notes in Computer Science, vol. 710, Springer, Berlin, 1993, pp. 406–417.
[17] Z. Róka, Automates cellulaires sur graphes de Cayley, Ph.D. Thesis, École Normale Supérieure de Lyon (1994).
[18] Z. Róka, The Firing Squad Synchronization Problem on Cayley graphs, in: Proceedings of MFCS'95, Lecture Notes in Computer Science, vol. 969, Springer, Berlin, 1995, pp. 402–411.