Simulations Between Cellular Automata on Cayley Graphs.

Zsuzsanna Roka

Laboratoire de l'Informatique du Parallélisme

Ecole Normale Supérieure de Lyon
Unité de recherche associée au CNRS n°1398

Simulations Between Cellular Automata on Cayley Graphs

Zsuzsanna Róka

Decembre 1994

Research Report N^{O} 94-40

Ecole Normale Supérieure de Lyon
46 Allée d'Italie, 69364 Lyon Cedex 07, France
Téléphone : (+33) 72.72.80.00   Télécopieur : (+33) 72.72.80.80
Adresse électronique : lip@lip.ens−lyon.fr

Simulations Between Cellular Automata on Cayley Graphs

Zsuzsanna Róka

Decembre 1994

Abstract

We consider cellular automata on Cayley graphs and compare their computational powers according to the architecture on which they work. We show that, if there exists a homomorphism with a finite kernel from a group into another one such that the image of the first group has a finite index in the second one, then every cellular automaton on the Cayley graph of one of these groups can be uniformally simulated by a cellular automaton on the Cayley graph of the other one. This simulation can be constructed in a linear time. With the help of this result we also show that cellular automata working on any Archimedean tiling can be simulated by a cellular automaton on the grid of $\mathbb{Z}^2$ and conversely.

Keywords: cellular automata, Cayley graphs, simulations

Résumé

Nous comparons la puissance de calcul des automates cellulaires agissant sur différents graphes de Cayley. Nous montrons que, s'il existe un morphisme à noyau fini d'un groupe dans un autre tel que l'indice de l'image du premier groupe est fini dans le deuxième, alors tout automate cellulaire sur le graphe de Cayley d'un de ces groupes peut-être simulé par un automate cellulaire sur le graphe de Cayley de l'autre groupe avec un facteur de perte de temps linéaire. Nous montrons aussi, que les automates cellulaires agissant sur les pavages Archimédiens peuvent être simulés par un automate cellulaire sur la grille de $\mathbb{Z}^2$ et réciproquement.

Mots-clés: automates cellulaires, graphes de Cayley, simulations

Simulations Between Cellular Automata on Cayley Graphs*

Zsuzsanna Róka

Laboratoire de l'Informatique du Parallélisme
ENS-Lyon, 46 Allée d'Italie
69364 Lyon Cedex 07
France
zroka@lip.ens-lyon.fr

1 Introduction

A cellular automaton (CA) is a network of identical finite automata which work in parallel and synchronously. It is also required that the network be regular, thus, it can be considered as a Cayley graph of a finitely presented group. Automata are placed on the vertices of the graph, and they communicate with each other through the edges. Some recent papers [5, 8, 6] have already considered this generalized notion of cellular automata, as we do in this paper.

Our goal is to compare the computational power of different models, more precisely, the power of cellular automata working on different Cayley graphs. In order to do that, we study simulations between them. The notion of a simulation is very intuitive but has never been studied for itself. In [7], we have shown with the help of various examples that this notion is very complicated, and we have given some possible definitions for it. Here we shall use a simplified definition which is not the most general one, but fits every simulation presented in our paper.

First, we study some examples: we construct simulations between cellular automata with von Neumann, hexagonal, triangular neighborhoods and also cellular automata on trees. These examples allow us to give a sufficient condition for every cellular automaton on a Cayley graph to be simulated by a

*This work was partially supported by the Esprit Basic Research Action "Algebraic and Syntactical Methods In Computer Science", by the PRC "Mathématique et Informatique" and by the grant of the French Government.

cellular automaton on another Cayley graph. We show that this condition is not necessary. When it holds, simulations are rather simple.

We remark that the same underlying graph can be colored in several ways corresponding to basically different groups. We present a more complicated simulation between hexagonal cellular automata on two different Cayley graphs. Then, we give a sufficient condition for a simulation to be possible in both directions between cellular automata.

In the last section, we show that every planar, modular structure is equivalent to the grid $\mathbb{Z}^2$ with respect to linear time simulations.

2 Definitions

In this section, we recall some algebraic notions in order to define cellular automata on Cayley graphs.

2.1 Presentation of a group

Let $\boldsymbol{G}$ be a group and X its element set, and let $G = \{g_1, g_2, \ldots\}$ (possibly infinite set) be a subset of X. We denote by G^{-1} the set of the inverse elements of G: $G^{-1} = \{g_1^{-1}, g_2^{-1}, \ldots\}$. If we consider the free monoid on $G \cup G^{-1}$, that is, the set of words on $G \cup G^{-1}$, we can associate to a word w an element $[w]$ of $\boldsymbol{G}$. More than one words can correspond to one element of $\boldsymbol{G}$. If every element of $\boldsymbol{G}$ can be expressed as a word on $G \cup G^{-1}$, we say that G is a *generating set* for $\boldsymbol{G}$. We define a *relation* as an equality between two words in $\boldsymbol{G}$. A generator g is said to be *idempotent* if $g^2 = 1$. If $\boldsymbol{G}$ is generated by $G = \{g_1, g_2, \ldots\}$ and if every relation in $\boldsymbol{G}$ can be deduced from relations $R = \{p = p',\ q = q',\ r = r',\ \ldots\}$, then we write

$$\boldsymbol{G} = \langle g_1, g_2, \ldots \mid p = p', q = q', r = r', \ldots \rangle \quad (\boldsymbol{G} = \langle G \mid R \rangle)$$

and $\langle g_1, g_2, \ldots \mid p = p', q = q', r = r', \ldots \rangle$ is said to be a *presentation of* $\boldsymbol{G}$. A presentation is said to be *finitely generated* (*finitely related*) if the number of generators (defining relations) is finite. A *finite presentation* is both finitely generated and finitely related.

In this paper, we shall only study finitely presented infinite groups.

2.2 Cayley graphs

For every group presentation $\boldsymbol{G} = \langle G \mid R \rangle$ there is an associated *Cayley graph* $\Gamma = (V, A)$: the vertices (V) correspond to the elements of the group, and the arcs (A) are colored with 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 $\boldsymbol{G}$. Remark that the Cayley graph depends on the group presentation and not on the group itself.

Remark 1 *From now on, if we refer to a group $\boldsymbol{G} = \langle G \mid R \rangle$, we refer to its presentation and not to the group itself. Thus, we shall talk about cellular automaton defined on the Cayley graph of a group $\boldsymbol{G} = \langle G \mid R \rangle$ and not cellular automaton defined on the Cayley graph of the presentation $\langle G \mid R \rangle$ of group $\boldsymbol{G}$.*

The following properties hold in Cayley graphs Γ:

Property 1 (*Cayley graphs* 1): In Γ the arcs have a regular coloring with the generators: for each vertex v and generator g, there exists exactly one arc colored with g starting at v, and exactly one arc colored with g ending at v.

If, in a group, the relation $g = g'$ holds for two generators g and g', we delete one of them from the generating set and replace all of its occurrences by the other one in all relations. Thus, we have also the following property:

Property 2 (*Cayley graphs* 2): If there exists an arc colored with g from the vertex i to the vertex j, then it is the only one from i to j.

Remark that the same undirected graph can sometimes be colored in several ways. See an example for the graph which gives the triangular tiling of the plane. There are seven possible colorings, here we present only two of them: the Cayley graphs of groups $\boldsymbol{G} = \langle a, b, c \mid abc = 1, ab = ba \rangle$ and $\boldsymbol{G}' = \langle a, b, c \mid a^3 = 1, b^3 = 1, c^3 = 1, cba = 1 \rangle$ are shown in Figure 1a, 1 b (the other colorings are presented in Section 7).

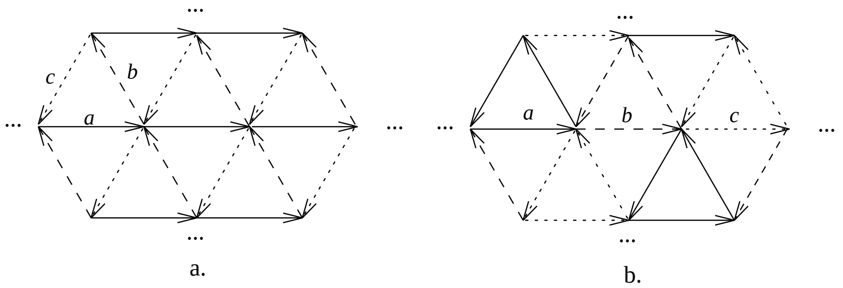


Figure 1: Two colorings for triangular tiling.

2.3 Cellular automata on Cayley graphs

As Cayley graphs are graphical representations of groups, they have very regular structures. Hence, by putting automata in the vertices, we can obtain a general notion of cellular automata in an analog way as A. Machí and F. Mignosi in [5].

Definition 1 *A* cellular automaton on a Cayley graph $\Gamma = (V, A)$ *is a 4-tuple* $\mathcal{A} = (S, \Gamma, N, \delta)$ *where*

- S *is a finite set, called the* set of states*,*
- Γ *is the* Cayley graph *of a finitely presented group* $\boldsymbol{G} = \langle G \mid R \rangle$*,*
- *The* neighborhood N *is a vector defined by words of* $\boldsymbol{G}$*:* $N = (w_1, w_2, \ldots, w_m)$ *where* $\forall i, w_i$ *is in* $G \cup G^{-1} \cup \{1\}$*,*
- $\delta : S^m \to S$ *is the* local transition rule

In an analog way as for cellular automata in $\mathbb{Z}^n$, we can characterize the global behavior of the cellular automaton. A *configuration* is an application c from $\boldsymbol{G}$ to S, so the set C of all configurations is $S^{\boldsymbol{G}}$. A cellular automaton transforms a configuration into another one:

$$\forall c \in C,\ \forall i \in \boldsymbol{G},\ \mathcal{A}(c)(i) = \delta(c(iw_1), c(iw_2) \ldots, c(iw_m)).$$

Remark that a more general definition of a cellular automaton can also be done by defining the neighborhood with words on $G \cup G^{-1}$, and not only with generators and their inverses. However, we have shown in [7] that we do not loose any of the generality of the definition for cellular automata if we make this restriction. Moreover, this definition allows us to consider that cells communicate through the arcs of the graph. For short, we shall call automaton (or cell) v the automaton put in vertex v of Γ.

2.4 The notion of a simulation between cellular automata.

Many papers have already studied various simulations between cellular automata, but the formal definition of a simulation has not been clearly given. In [7, 8, 6] we study this notion in details, and we show through some examples, why this notion is not easy to formally define and to work with. In this paper, we shall compare the computational power of cellular automata whose neighborhoods are complete, that is, they contain all generators, all inverse generators and also the neutral element. We design some simulations in the sense of the following definition:

Definition 2 *Let* $\mathcal{A} = (S, \Gamma, N, \delta)$ *be a cellular automaton and* $C_{\mathcal{A}}$ *the set of its configurations. Let* $\mathcal{B} = (S', \Gamma', N', \delta')$ *be a cellular automaton and* $C_{\mathcal{B}}$ *the set of its configurations. We say that* $\mathcal{B}$ simulates $\mathcal{A}$*, if there exist an injective application* $f : C_{\mathcal{A}} \to C_{\mathcal{B}}$ *and a constant* T *in* $\mathbb{N}$ *such that for all* c *in* $C_{\mathcal{A}}$

$$f(\mathcal{A}(c)) = \mathcal{B}^T(f(c)).$$

T is the simulation time factor, that is, the time which is necessary for $\mathcal{B}$ to simulate one iteration of $\mathcal{A}$. It depends on f but not on c. This definition can be illustrated by the following diagram:

$$\begin{tikzcd}
c \arrow[r, "f"] \arrow[d, Rightarrow, "\mathcal{A}"'] & f(c) \arrow[d, Rightarrow, "\mathcal{B}^T"] \\
\mathcal{A}(c) \arrow[r, "f"] & \mathcal{B}^T(f(c))
\end{tikzcd}$$

This definition does not cover all simulations. In [7] we have studied the problem of cellular automata with only one-way communications between cells: "given a Cayley graph, whether all bidirectional cellular automaton can be simulated by a one-way cellular automaton on this graph?" Sometimes, such a simulation is possible, but after the simulation of each iteration of the bidirectional cellular automaton, this configuration is "shifted" in the one-way cellular automaton. In order to understand this phenomenon, let us consider the example of the line, that is, the Cayley graph of the group $\boldsymbol{G} = \langle a \mid \emptyset \rangle$. The neighborhood of bidirectional cellular automata is given by $N = (a, a^{-1}, 1)$ and the neighborhood of one-way cellular automata is (for example) by $N = (a, 1)$. Let $\mathcal{A}$ be a bidirectional cellular automaton and $\mathcal{O}$ a one-way cellular automaton simulating $\mathcal{A}$. Let the initial configuration of $\mathcal{O}$ be the same as $\mathcal{A}$'s. It is clear that the transition of a cell v of $\mathcal{A}$ cannot be computed in the cell itself: it cannot know the state of its neighbor defined by a^{-1}. However, all needed states can arrive in cell va^{-1}. It means then, that the configuration of $\mathcal{A}$ after the first iteration can be found in $\mathcal{O}$, but with a "shift" a^{-1}. Simulations without shifts are not possible. The Definition 2 does not allow such simulations. However, all along this paper, we restrict ourselves to this definition, because we do not need shift when simulating: shifts are needed only in the case of cellular automata with one-way communications.

This definition, even if it is not the most general one, covers also very complicated simulations. In [7] we have shown with two examples that

- f is not necessarily a recursive function,
- a state of a simulated cellular automaton can be "splittered" in the simulating cellular automaton,
- more states of a simulated cellular automaton can be grouped in the simulating one.

In this paper we do not consider simulations where a state can be splittered: we consider that every state of a simulated cellular automaton is an "atomic" information. If every state in the initial configuration of the simulating cellular automaton depends on only one state of the simulated cellular automaton, we shall say that the simulation is *elementary* (ie. the states are not grouped).

First, we shall present some simulation results between cellular automata with some classical neighborhoods as von Neumann, hexagonal and triangular ones.

3 Some examples with classical neighborhoods

3.1 Von Neumann neighborhood

In the n-dimensional space $\mathbb{Z}^n$, the von Neumann neighborhood is defined with n-dimensional unit-vectors and their inverses. These unit-vectors form an independent vector system, so we can define them in a similar way with the help of Cayley graphs: the grid of $\mathbb{Z}^n$ can be colored as the Cayley graph of the Abelian group with a minimal generating set of n generators.

Definition 3 *An* n-dimensional von Neumann cellular (resp. one-way cellular) automaton *is a cellular (resp. one-way cellular) automaton on the Cayley graph of the group*

$$\boldsymbol{G}_{n\_VN} = \langle g_1, g_2, \ldots, g_n \mid g_i g_j = g_j g_i,\ 1 \leq i < j \leq n \rangle.$$

The Cayley graph of this group for the 2-dimensional case is shown in Figure 2.

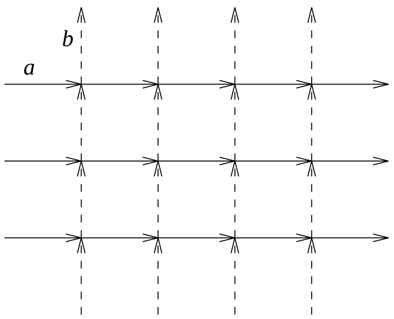


Figure 2: The Cayley graph of group $\boldsymbol{G} = \langle a, b \mid ab = ba \rangle$.

Here, we study, whether n-dimensional von Neumann CA can be simulated by m-dimensional von Neumann CA.

Definition 4 *Let* Γ *be the Cayley graph of group* $\boldsymbol{G}_{n\_VN}$. *We call* ball of radius k *and denote by* B_k^n *the set of all vertices being at distance at most* k *from 1. We denote by* $\#(B_k^n)$ *the number of vertices in* B_k^n.

Theorem 1 *For* $n > m$, n*-dimensional von Neumann CA cannot be simulated by* m*-dimensional von Neumann CA.*

Proof. We show this result by the simple fact that the growth of two Cayley graphs is different. We suppose that every n_VNCA can be simulated by a m_VNCA. Study the first iteration of the n_VNCA. In order to compute the transition of cell 1 for the n_VNCA, the states of ball B_1^n are needed. Let T be the simulation time factor. Then, these states must be found in the initial configuration of m_VNCA in the ball B_T^m. In order to simulate the second iteration of n_VNCA, the states of cells of ball B_2^n are necessary, hence they must be in the ball B_{2T}^m. The transition of a cell of n_VNCA at time t is computed in function of the states of cells being at distance at most t, hence the states of cells of ball B_t^n must be in the ball B_{tT}^m of m_VNCA, and so on. It means that if the simulation is elementary, then $\#(B_t^n) \leq \#(B_{tT}^m)$, for all t. If it is not elementary, then let k is the maximal number of states belonging to a cell of m_VNCA; then, we have $\#(B_t^n) \leq \#(B_{tTk}^m)$. As $\#(B_t^n) = O(t^n)$ and $\#(B_{tTk}^m) = O(t^m)$, if t is big enough, then $\#(B_t^n) > \#(B_{tTk}^m)$, which leads to a contradiction. □

As $\boldsymbol{G}_{m\_VN}$ is a subgroup of $\boldsymbol{G}_{n\_VN}$, if $n \geq m$, a simulation in the other direction is always possible without any loss of time (by "ignoring" the other dimensions). With an analog proof, we can show the following theorem.

Theorem 2 *Let B_k and B'_k be the balls of radius k in the Cayley graphs Γ and Γ', respectively. If $\#(B_k)$ is a polynomial of degree p and $\#(B'_k)$ is a polynomial of degree q, if $p \leq q$, then not every CA on Cayley graph Γ can be simulated by a CA on Cayley graph Γ'.*

This result is not very astonishing. S. Cole has been shown in [2], that the language recognition power of CA increases with the dimension of the space, which is a similar statement.

We shall now study simulations between CA with different kinds of neighborhoods.

3.2 Von Neumann and hexagonal neighborhoods

First of all, we present an intuitive definition of a hexagonal cellular automaton, then we define it formally, on Cayley graphs.

A *hexagonal cellular automaton* is usually defined as a cellular automaton in the plane $\mathbb{R}^2$, where the cells are at the centers of hexagons which tile the plane, and the neighbors of a cell are the cells located at the center of the adjacents hexagons (see Figure 3).

We present below a formal definition with the help of Cayley graphs.

Definition 5 *A* hexagonal cellular (one-way cellular) automaton *is a cellular (one-way cellular) automaton defined on the Cayley graph of the group*

$$\boldsymbol{G}_h = \langle a, b, c \mid ab = ba, abc = 1 \rangle$$

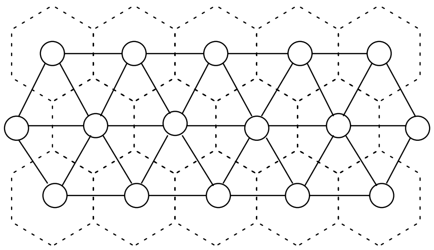


Figure 3: The hexagonal neighborhood.

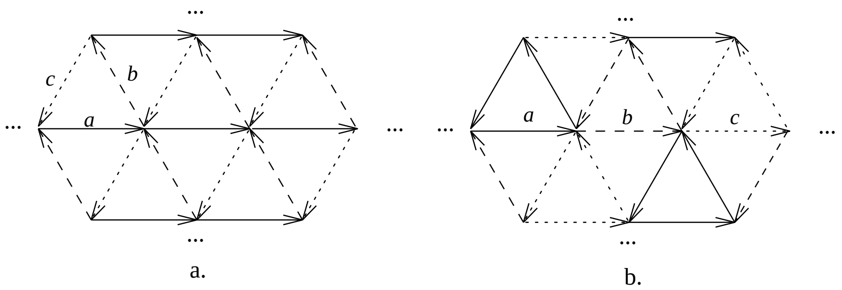


Figure 4: Cayley graphs of $\boldsymbol{G}_h$ and $\boldsymbol{G}_{h_2}$.

The Cayley graph of $\boldsymbol{G}_h$ is shown in Figure 4a.

Physicists have shown in [3] that, in certain cases, 3_VN nets can be simulated by hexagonal nets. The idea comes from the identical number of neighborhoods of a cell: 7 in both cases. Unfortunately, in general the simulation is not possible. Here, we give all possible simulations between CA with these two kinds of neighborhoods, and show when a simulation is not possible. For short, we denote hexagonal CA by HCA. In the plane, all given simulations are very simple, because these cellular automata are defined on the Cayley graphs of isomorphic groups. Later we give some more complicated examples.

Proposition 1 *For $n \geq 2$, every HCA can be simulated by an n_VNCA.*

Proof. First we prove this proposition for $n = 2$: let $\mathcal{A} = (S_h, \Gamma_h, N_h, \delta_h)$ be a hexagonal CA defined by Definition 5 with an initial configuration $c^0_{\mathcal{A}}$.
We construct a 2_VNCA $\mathcal{B} = (S_{2\_VN}, \Gamma_{2\_VN}, N_{2\_VN}, \delta_{2\_VN})$ which simulates $\mathcal{A}$. First, let us consider the elements of $\boldsymbol{G}_h$ as words in $\{a, b\}$. We can do this, because $c = a^{-1}b^{-1}$. Let $\gamma : \boldsymbol{G}_h \to \boldsymbol{G}_{2\_VN}$ be defined by $\gamma(u) = u$ for all $u \in \boldsymbol{G}_h$. Let S_{2_VN} be a sup-set of S_h. We define the initial configuration $c^0_{\mathcal{B}}$ of $\mathcal{B}$ by $c^0_{\mathcal{B}}(\gamma(u)) = c^0_{\mathcal{A}}(u)$ for all u in $\boldsymbol{G}_h$, see Figure 5.

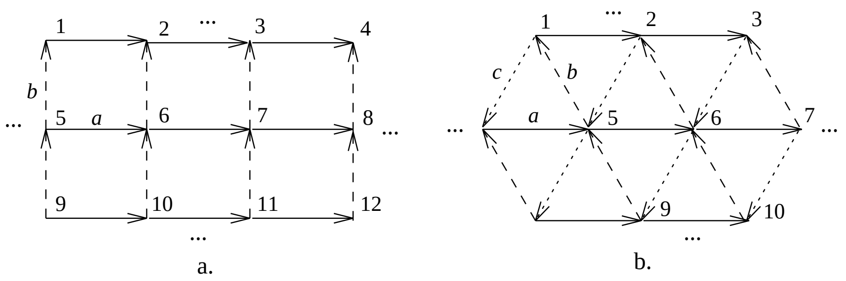


Figure 5: Initial configurations of 2D_VNCA and HCA.

For computing the transition of a cell v of the HCA, the states of all of its neighbors defined by a, b, c, a^{-1}, b^{-1}, c^{-1} are needed. In $c^0_{\mathcal{B}}$, they can be found in cells defined by va, vb, $va^{-1}b^{-1}$, va^{-1}, vb^{-1} and vab, respectively. As not all of these cells are neighbors of v by the von Neumann neighborhood, the simulation cannot be given with $T = 1$. They are in a ball of radius 2, so $T = 2$ is possible. This simulation can be done as in all previous simulations: the first step of the simulation of an iteration is a memorizing step, and in the second step cells can compute the transitions of $\mathcal{A}$. Formally, we define $\mathcal{B}$ by

$$\begin{gathered}
S_{2\_VN} = S_h \cup S_h^5 \\
\delta_{2\_VN} : S^5_{2\_VN} \rightarrow S_{2\_VN} \\
\delta_{2\_VN}(x, y, z, r, s) = (x, y, z, r, s) \\
\delta_{2\_VN}((x_1, x_2, \ldots, x_5), (y_1, y_2, \ldots, y_5), (z_1, z_2, \ldots, z_5), (r_1, r_2, \ldots, r_5), \\
(s_1, s_2, \ldots, s_5)) = \delta_h(x_1, x_2, x_3, x_4, r_4, y_2, x_5).
\end{gathered}$$

For $n = 3$: as $\boldsymbol{G}_{2\_VN} \subset \boldsymbol{G}_{3\_VN}$, we can define the simulation as before (by "ignoring" the third dimension): we define the initial configuration of $\mathcal{B}$ with $c^0_{\mathcal{B}}(u) = \omega$, $\omega \notin S_h$, for all u not being an image by Γ, and with a little modification of the transition function.

We give another simulation: we can use the third dimension in order to decrease the simulation time factor down to $T = 1$. We construct a 3_VNCA $\mathcal{B}$ simulating $\mathcal{A}$ as follows.

Let $\gamma : \boldsymbol{G}_h \rightarrow \mathcal{P}(\boldsymbol{G}_{3\_VN})$ be defined by

$$\begin{gathered}
\gamma(1) = \{1, abc, (abc)^{-1}, (abc)^2, (abc)^{-2}, \ldots\} \\
\gamma(u) = \{u, uabc, u(abc)^{-1}, u(abc)^2, u(abc)^{-2}, \ldots\},\ \forall u \in \boldsymbol{G}_h
\end{gathered}$$

Remark that each element of $\boldsymbol{G}_h$ has an infinite number of images by Γ, which was not the case in the previous simulations.

We define S_{3_VN} as a sup-set of S_h; the initial configuration $c^0_{\mathcal{B}}$ of $\mathcal{B}$ is defined by $c^0_{\mathcal{B}}(\gamma(u)) = c^0_{\mathcal{A}}(u)$ and is shown in Figure 6b. With this construction, if two

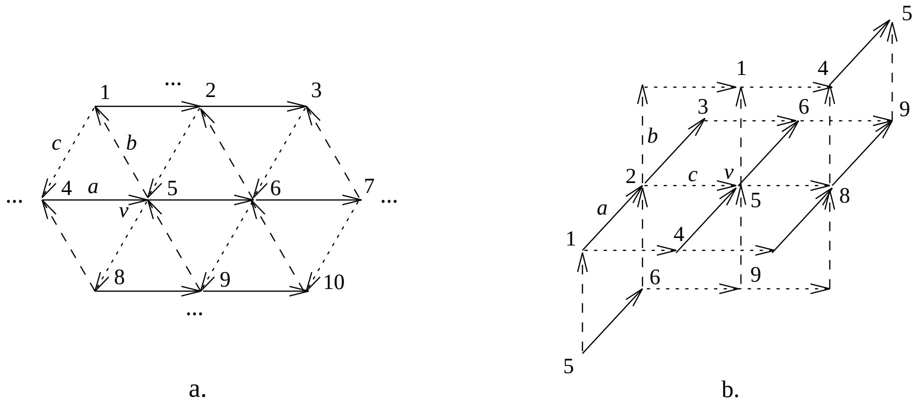


Figure 6: Initial configurations of HCA and 3_VNCA.

information are neighbors in $c^0_{\mathcal{A}}$, they are also neighbors in $c^0_{\mathcal{B}}$. So $\mathcal{B}$ can simply defined by

$$S_{3\_VN} = S_h$$
$$\delta_{3\_VN} : S^7_{3\_VN} \rightarrow S_{3\_VN}$$
$$\delta_{3\_VN}(x, y, z, r, s, t, u) = \delta_h(x, y, z, r, s, t, u)$$

For $n > 3$, as Γ_{3_VN} is a subgraph of Γ_{n_VN} and every CA on Γ_{3_VN} can be simulated by a CA on Γ_{n_VN} with $T = 1$, hence every HCA can be simulated also by a n_VNCA $n > 2$, with a simulation time factor $T = 1$. □

Proposition 2 *Every 2_VNCA can be simulated by a HCA.*

Proof. The simulation can be given as in the converse direction; the von Neumann neighborhood is included in the hexagonal neighborhood, so for computing the transition of a cell of the 2_VNCA in the HCA, it is sufficient to choose the needed information.

Formally, let $\mathcal{A} = (S_{2\_VN}, \Gamma_{2\_VN}, N_{2\_VN}, \delta_{2\_VN})$ be a 2_VNCA. We construct a hexagonal CA $\mathcal{B} = (S_h, \Gamma_h, N_h, \delta_h)$ in a "natural" way which simulates it with $T = 1$.
Let $\gamma : \boldsymbol{G}_{2\_VN} \rightarrow \boldsymbol{G}_h$ be an application defined by $\gamma(u) = u$ for all $u \in \boldsymbol{G}_{2\_VN}$. Let $S_h = S_{2\_VN}$. We define the initial configuration $c^0_{\mathcal{B}}$ of $\mathcal{B}$ by $c^0_{\mathcal{B}}(\gamma(u)) = c^0_{\mathcal{A}}(u)$. We can see, that neighbor information in $c^0_{\mathcal{A}}$ are also neighbors in $c^0_{\mathcal{B}}$, hence we can define $\mathcal{A}$ by

$$S_h = S_{2\_VN}$$
$$\delta_h(x, y, z, r, s, t, u) = \delta_{2\_VN}(x, y, r, s, u)$$

□

Theorem 3 *For $n > 2$, n_VNCA cannot be simulated by HCA.*

Proof. Hexagonal CA can be simulated by 2_VNCA, and conversely. If n_VNCA could be simulated by HCA, then it would imply that n_VNCA can be simulated by 2_VNCA which is in contradiction with Theorem 1. □

In Section 2.2 we have remarked that the same non-oriented graph can be colored in different ways. We have also shown in Figure 1 the Cayley graph of group $\boldsymbol{G}_{h_2} = \langle a, b, c \mid a^3 = 1, b^3 = 1, c^3 = 1, cba = 1 \rangle$. Its underlying graph gives the same triangular tiling of the plane, so we could have use it to define hexagonal. Similar simulation results can be done but these simulations are a bit more complicated, because $\boldsymbol{G}_{2\_VN}$ and $\boldsymbol{G}_{h_2}$ are not isomorphic, while $\boldsymbol{G}_{2\_VN}$ and $\boldsymbol{G}_h$ are. This means that not only the physical architecture of cellular automata is important for simulations, but also the local communications, that is, Cayley graphs on which we define them. We shall study this problem in Section 5.1.

3.3 Simulations between CA on the Cayley graph of free-groups

Here we study a bit more complicated simulations. First of all, we study the Cayley graphs Γ and Γ' of the groups $\boldsymbol{FR}_3 = \langle a, b, c \mid \emptyset \rangle$ and $\boldsymbol{FR}_2 = \langle a, b \mid \emptyset \rangle$, respectively (see Figure 7).

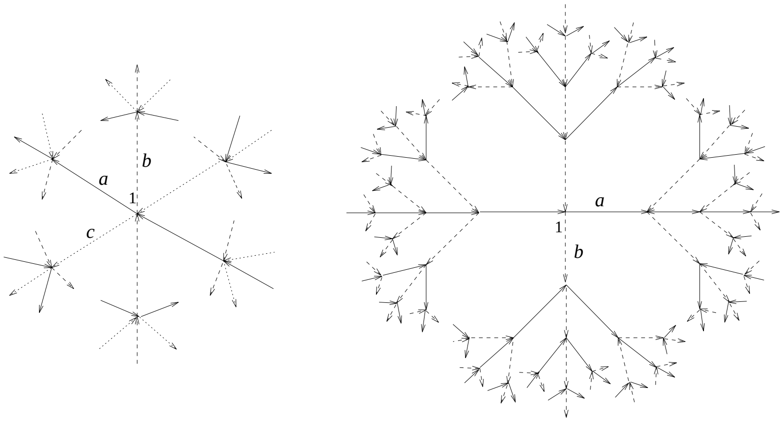


Figure 7: The Cayley graphs of $\boldsymbol{FR}_3$ and $\boldsymbol{FR}_2$.

Let $\mathcal{A} = (S, \Gamma, N, \delta)$ be a cellular automaton. We want to construct a cellular automaton $\mathcal{B} = (S', \Gamma', N', \delta')$ which simulates $\mathcal{A}$. Let $\gamma : \boldsymbol{FR}_3 \rightarrow \boldsymbol{FR}_2$ be a homomorphism defined by

$$\begin{array}{llll}\gamma(1)=1 & \gamma(a)=ab & \gamma(b)=ba & \gamma(c)=a^2\\ \gamma(uv)=\gamma(u)\gamma(v), & u,v\in \boldsymbol{FR}_3. & \text{(see Figure 8)} & \end{array}$$

We define S' as a sup-set of S. Let the initial configuration of $\mathcal{B}$ given by

$$c^0_{\mathcal{B}}(\gamma(u)) = c^0_{\mathcal{A}}(u),\ u\in \boldsymbol{FR}_3.$$

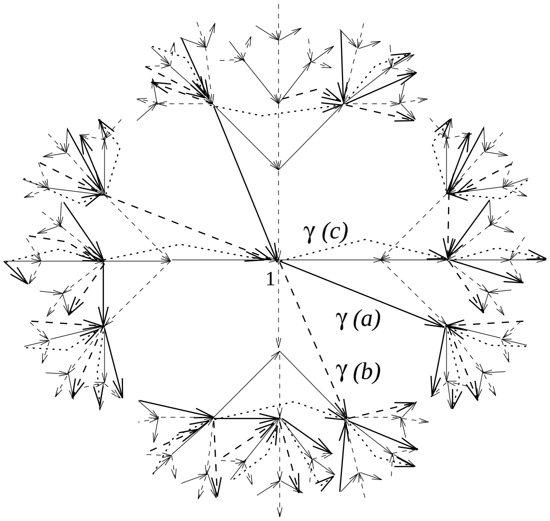


Figure 8: The mapping $\gamma : \boldsymbol{FR}_3 \rightarrow \boldsymbol{FR}_2$

Then, $\mathcal{B}$ is defined by

$$\begin{gathered} S' = S \cup S^5 \\ \delta' : S'^5 \rightarrow S' \\ \delta'(x,y,z,r,s) = (x,y,z,r,s) \\ \delta'((x_1,\ldots,x_5),(y_1,\ldots,y_5),(z_1,\ldots,z_5),(r_1,\ldots,r_5),(s_1,\ldots,s_5)) = \\ \delta(x_2,y_1,x_1,r_3,z_4,z_3,s_5). \end{gathered}$$

With this construction, neighbor states in $\mathcal{A}$ are not neighbors in $\mathcal{B}$: while the needed states to compute the new state of a cell arrive by single arcs in $\mathcal{A}$, they arrive by pairs of arcs in $\mathcal{B}$. So, in the first step of the simulation of an iteration, all cells store the states of all of its neighbors, and in the second step they compute the transition of $\mathcal{A}$ (the simulation time factor is $T = 2$).

Let us now study free groups generated by any number of generators.

Theorem 4 *Every CA $\mathcal{A}$ defined on the Cayley graph of a free-group with n generators $\boldsymbol{FR}_n$ can be simulated by a CA $\mathcal{B}$ defined on the Cayley graph of another free-group with $m(>1)$ generators $\boldsymbol{FR}_m$ with a simulation time factor $\lceil \log_m n \rceil$.*

Proof. The assertion is true if $m \geq n$, because we can define the initial configuration of a simulating CA by "ignoring" some branches of the tree. If $m < n$, then we define a mapping $\gamma : \boldsymbol{FR}_n \rightarrow \boldsymbol{FR}_m$ with

$$\begin{gathered}\gamma(g_1) = w_1 \\ \gamma(g_2) = w_2 \\ \vdots \\ \gamma(g_n) = w_n \\ \gamma(uv) = \gamma(u)\gamma(v), \;\; u, v \in \boldsymbol{FR}_n\end{gathered}$$

where $w_1, \ldots, w_n$ are different words of the same length k. If $k \geq \lceil \log_m n \rceil$, then these w_i's can be given. We can construct a simulation with a simulation time factor k in a similar way as in the case of CA on the Cayley graph of $\boldsymbol{FR}_3$ and $\boldsymbol{FR}_2$. □

3.4 An example for simulation by grouping states

Sometimes, only non-elementary simulations are possible between two cellular automata. Here we give an example for such a simulation.

Example 1 (*Cylinder automaton on the line*):

Let $\boldsymbol{G} = \langle a, b \mid ab = ba, b^2 = 1 \rangle$ and $\boldsymbol{G}' = \langle a \mid \emptyset \rangle$ and Γ, Γ' their Cayley graphs, respectively. Let $\mathcal{A} = (S, \Gamma, N, \delta)$ be a CA. Let $\gamma : \boldsymbol{G} \rightarrow \boldsymbol{G}'$ be a homomorphism defined by $\gamma(1) = 1$, $\gamma(b) = 1$, $\gamma(a) = a$ and for all u, for all v in $\boldsymbol{G}$, $\gamma(uv) = \gamma(u)\gamma(v)$ (see Figure 9). We build a CA $\mathcal{B} = (S', \Gamma', N', \delta')$ simulating $\mathcal{A}$ by

$$\begin{gathered}S' = S \cup S^2 \\ \delta' : (S')^3 \rightarrow S' \\ \delta'((x_1, x_2), (y_1, y_2), (z_1, z_2)) = (\delta(z_2, x_2, z_1), \delta(x_1, z_1, x_2))\end{gathered}$$

starting from the initial configuration given by $c^0_{\mathcal{B}}(\gamma(u)) = c^0_{\mathcal{A}}(u)$ for all u in $\boldsymbol{G}$.

4 A sufficient condition

In the simulation between CA defined on the Cayley graph of free groups, we can remark that γ is an injective homomorphism. We can then remark also that in Example 1, γ is a homomorphism with a finite kernel $\{1, b\}$. In general we can state that:

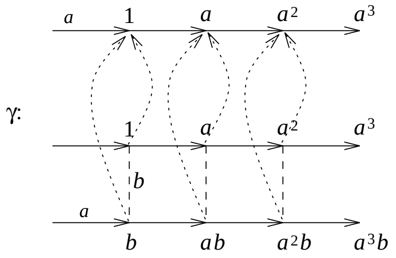


Figure 9: The homomorphism $\gamma : \langle a, b \mid ab = ba, b^2 = 1\rangle \rightarrow \langle a \mid \emptyset\rangle$.

Theorem 5 *If there exists a homomorphism γ with a finite kernel from a group $\boldsymbol{G}$ into another group $\boldsymbol{G}'$, then every cellular automaton defined on the Cayley graph of $\boldsymbol{G}$ can be simulated by a cellular automaton defined on the Cayley graph of $\boldsymbol{G}'$.*

Proof. Let $\mathcal{A} = (S, \Gamma, N, \delta)$ be a cellular automaton and $\gamma : \boldsymbol{G} \rightarrow \boldsymbol{G}'$ a homomorphism with a finite kernel: for each generator g_i in $\boldsymbol{G}$, $\gamma(g_i) = w_i$ where for all i, w_i is a word in $\boldsymbol{G}'$ and $\max\{|w_i|\} = m$. We want to construct a cellular automaton $\mathcal{B} = (S', \Gamma', N', \delta')$ which simulates $\mathcal{A}$. First, let γ be injective. Let n be the number of neighbors of a cell in $\mathcal{B}$. Let ω be a state not belonging to S. We define the set of states of $\mathcal{B}$ by

$$S' = (S \cup \{\omega\}) \cup (S \cup \{\omega\})^n \cup (S \cup \{\omega\})^{n^2} \cup \ldots \cup (S \cup \{\omega\})^{n^m}$$

and the transition function by

$$\delta' : S'^n \rightarrow S'$$

in the following way. Cells with no preimage are in state ω. At time 1, they store all states of all of their neighbors at time 0:

$$\delta'(x_1, x_2, \ldots, x_n) = (x_1, x_2, \ldots, x_n).$$

At time 2, they store all states of all of their neighbors at time 1, that is, the states of those cells at time 0, which are at distance at most 2 from the cell:

$$\delta'((x_{11}, x_{12}, \ldots, x_{1n}), \ldots, (x_{n1}, x_{n2}, \ldots, x_{nn})) = (x_{11}, x_{12}, \ldots, x_{nn}).$$

At time m, cell v stores the states of all cells which are defined by generators, inverse generators, words of length two, words of length 3, ..., words of length m in G'. As every w_i is a word of length at most m, its state is known by the cell v. As the local function is the same for each cell and γ is a homomorphism, for every cell v in $\mathcal{B}$, for all i, the state of the cell vw_i arrives by the same path and as the same component of vector, hence, at time m, the transitions of $\mathcal{A}$ can be computed in $\mathcal{B}$.

We now study the case where $\gamma : \boldsymbol{G} \rightarrow \boldsymbol{G}'$ is not injective but its kernel is finite. As the kernel of a homomorphism forms a group, this kind of simulation is possible only if $\boldsymbol{G}$ has a finitely presented non-trivial finite subgroup. The simulation can be defined as before, the only difference is that, while in the previous case, in each cell of $\mathcal{B}$, the transition of a single cell of $\mathcal{A}$ is computed, here the transitions of all cells which have the same image by γ are computed. □

Remark that if γ is injective, the simulation constructed in this way is elementary.

Corollary 1 *If two groups are isomorphic, then cellular automata defined on their Cayley graphs can be simulated by each other in an elementary way.*

We cannot say anything about the simulation time factor, it depends on the Cayley graph.

5 Elementary simulations

In this section, we study only elementary simulations. We have seen, that the existence of an injective homomorphism allows elementary simulations. On the other hand, if there did not exist homomorphism with a finite kernel from a group into another one, then we could not define any simulation. In the following example we show, that in some cases, there does not exist a homomorphism with a finite kernel, but the simulation (even elementary) is possible: the condition given in Theorem 5 is not necessary. In Section 6, we study other, non-elementary simulations.

5.1 Hexagonal and triangular neighborhoods

First, we present the intuitive definition of a triangular cellular automaton, then we define it formally, on Cayley graphs.

A *triangular cellular automaton* is usually defined as a cellular automaton in the plane $\mathbb{R}^2$, where the cells are at the center of equilateral triangles, and the neighbors of a cell are the cells located at the center of the triangles which are adjacent side by side (see Figure 10).

Definition 6 *A* triangular cellular automaton *is a cellular automaton defined on the Cayley graph of the group*

$$\boldsymbol{G}_t = \langle a, b, c \mid a^2 = 1, b^2 = 1, c^2 = 1, (abc)^2 = 1 \rangle.$$

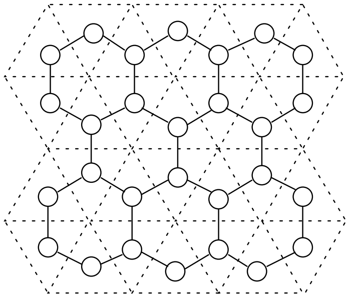


Figure 10: Triangular neighborhood.

In the Cayley graph of $\boldsymbol{G}_t$, as for all generator g in $\boldsymbol{G}_t$, $g^2 = 1$, between every pair of neighbor vertices there are two arcs colored with g: we replace them by single, non-oriented edges colored with g.

This definition is a bit special relatively to all the definitions we have given before: the neighborhood of a cell formally consists of 7 neighbors ($N = (a, b, c, a^{-1}, b^{-1}, c^{-1}, 1)$). In reality, it consists of only 4 neighbors, because each neighbor defined by a generator g is the same cell as the neighbor defined by the inverse generator g^{-1}.

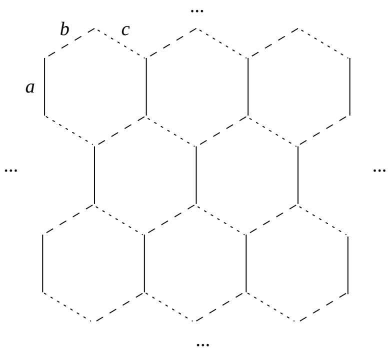


Figure 11: The Cayley graph of $\boldsymbol{G}_t$.

The definition of hexagonal cellular automata had already been given in Section 3.2 (Definition 5). We recall that they are defined on the Cayley graph of the group $\boldsymbol{G}_h = \langle a, b \mid ab = ba, abc = 1\rangle$.

We shall show that an elementary simulation can also be defined sometimes without the existence of any homomorphism with finite kernel. We shall denote triangular cellular automata by TCA.

Lemma 1 *There does not exist a homomorphism with a finite kernel from* $\boldsymbol{G}_t$ *to* $\boldsymbol{G}_h$*.*

Proof. We suppose that $\beta : \boldsymbol{G}_t \rightarrow \boldsymbol{G}_h$ is a homomorphism with a finite kernel, $\beta(a) = w_1$, $\beta(b) = w_2$ and $\beta(c) = w_3$. If $w_1 = w_2 = w_3 = 1$, then the kernel of β is infinite. Hence, at most one of the w_i's must be different from 1. We suppose that it is w_1. As w_1 is an element of $\boldsymbol{G}_h$, and $\boldsymbol{G}_h$ is commutative, w_1 can be expressed as $w_1 = a^n b^m \neq 1$. Then, $1 = \beta(1) = \beta(a^2) = \beta(a)\beta(a) = w_1^2 = a^{2n}b^{2m} \neq 1$, which leads to a contradiction. □

Lemma 2 *There exists an injective homomorphism from* $\boldsymbol{G}_h$ *to* $\boldsymbol{G}_t$*.*

Proof. Let $\beta : \boldsymbol{G}_h \rightarrow \boldsymbol{G}_t$ defined by $\beta(1) = 1$, $\beta(a) = ba$, $\beta(b) = ac$, $\beta(c) = cb$ and for all u and for all v in $\boldsymbol{G}_h$, $\beta(uv) = \beta(u)\beta(v)$. See Figure 12.

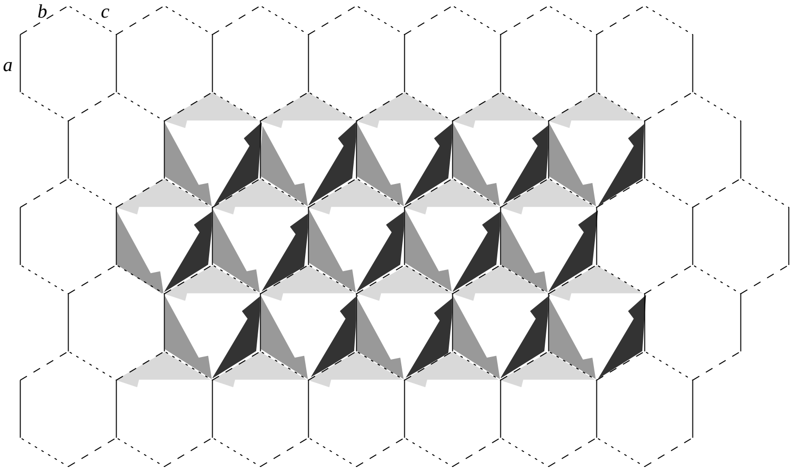


Figure 12: The injective homomorphism $\beta : \boldsymbol{G}_h \rightarrow \boldsymbol{G}_t$.

In order to show that β is a homomorphism, it is sufficient to show that $\beta(ab) = \beta(ba)$ and $\beta(abc) = 1$:

$$\begin{gathered}\beta(ab) = \beta(a)\beta(b) = ba.ac = bc \\ \beta(ba) = \beta(b)\beta(a) = ac.ba = ac.ba.abc.abc = bc \\ \beta(abc) = ba.ac.cb = 1\end{gathered}$$

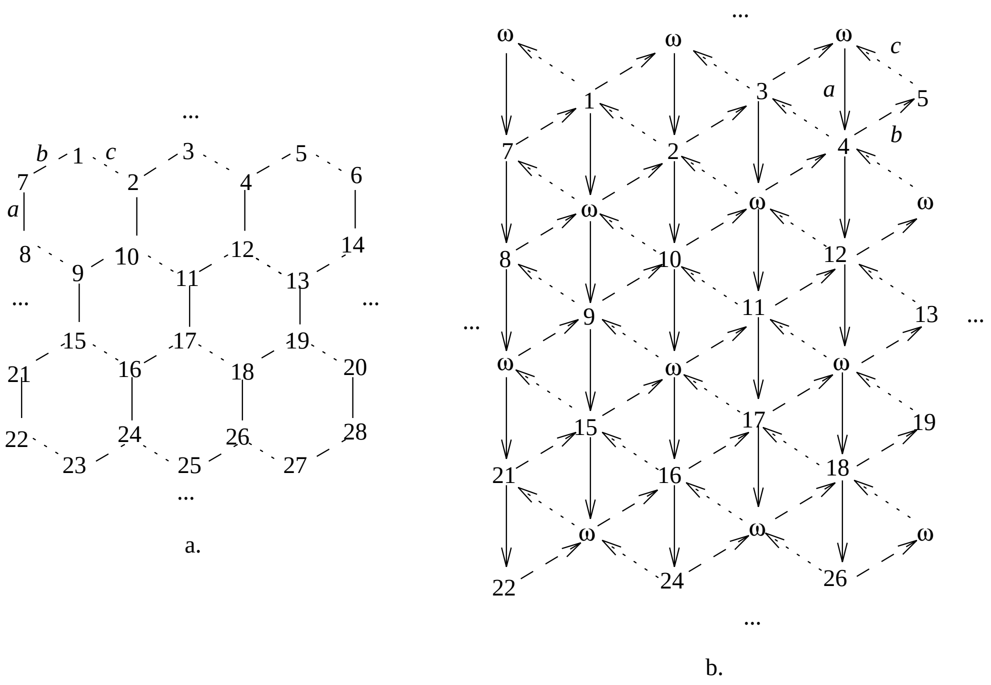
It is clear that β is an injective homomorphism. □

Proposition 3 *Every TCA can be simulated by a HCA in an elementary way, and conversely.*

Proof. Lemma 2 and Theorem 5 imply that every hexagonal CA can be simulated by a triangular CA.

Let us study now the converse simulation, when we want to simulate every TCA $\mathcal{T}$ with a HCA $\mathcal{H}$. We define the set of states of $\mathcal{H}$ by
$S_h = S_t \cup \{\omega\}$ $(\omega \notin S_t)$. The initial configurations of $\mathcal{T}$ and $\mathcal{H}$ are shown in Figure 13, cells without pre-image are in state ω. The transition function of $\mathcal{H}$ is given by

$$\begin{gathered}\delta_h(x,y,z,r,s,t,u) = \delta_t(x,y,z,u) \text{ if } s=t=u=\omega\\ \delta_h(x,y,z,r,s,t,u) = \delta_t(r,s,t,u) \text{ if } x=y=z=\omega\\ \delta_h(x,y,z,r,s,t,u) = \omega \text{ if } u=\omega.\end{gathered}$$



a.

b.

Figure 13: Initial configurations of TCA and HCA.

This construction is a bit different from the others. For example, consider the cell having information 12 in the initial configuration of $\mathcal{T}$. The states of its neighbor defined by the generator a (resp. b, c) (number 4 (resp. 11,13)) is in a neighbor cell defined by generator a^{-1} (resp. b^{-1}, c^{-1}) in the initial configuration of $\mathcal{H}$. Let us study now the cell numerated 11 in the initial configuration of $\mathcal{T}$. The state of its neighbor defined by the generator a (resp. b, c) is in a neighbor cell defined by a (resp. b, c) in the initial configuration of $\mathcal{H}$. So there are two types of cells, but we can define the transition function without contradiction, because if the needed states are in neighbors defined by a, b, c (resp. a^{-1}, b^{-1}, c^{-1}), then the others are in state ω (only one choice is available). □

As we have already noticed, the same graph can be colored in different ways. Here, we do not give all simulations between cellular automata on the other Cayley graphs, we study this problem in a more general way in Section 7. We take only one example. In Figure 4b, we show the Cayley graph of group

$$\boldsymbol{G}_{h_2} = \langle a, b, c \mid a^3 = 1, b^3 = 1, c^3 = 1, cba = 1 \rangle.$$

As this graph gives also the triangular tiling of the plane, it can be used in order to define hexagonal cellular automata. For short, we denote cellular automata on the Cayley graph of $\boldsymbol{G}_{h_2}$ by HCA2. In order to show that every HCA can be simulated by a HCA2, it is sufficient to give an injective homomorphism from $\boldsymbol{G}_h$ to $\boldsymbol{G}_{h_2}$.

Lemma 3 *There exists an injective homomorphism from $\boldsymbol{G}_h$ to $\boldsymbol{G}_{h_2}$.*

Proof. Let $\gamma : \boldsymbol{G}_h \to \boldsymbol{G}_{h_2}$ defined by

$$\gamma(a) = abc; \;\; \gamma(b) = cab; \;\; \gamma(c) = bca$$

and for all u, for all v in $\boldsymbol{G}_h$, $\gamma(uv) = \gamma(u)\gamma(v)$, see Figure 14.

In order to prove that γ is an injective homomorphism, it is sufficient to show that $\gamma(abc) = 1$ and $\gamma(ab) = \gamma(ba)$:

$\gamma(abc) = abc.cab.bca = ab.(cc).a.(bb).ca = ab.(c^{-1}).a.(b^{-1}).ca = ab.b(a.a.a)c.ca = a.(bb).(cc).a = a.(b^{-1}.c^{-1}).a = aaa = 1.$

$\gamma(ab) = abc.cab = ab.(c^{-1}).ab = a(b.b)(a.a)b = a(b^{-1})(a^{-1})b = (a.a)(c.c)(b.b) = a^{-1}c^{-1}b^{-1}$

$\gamma(ba) = cab.abc = ca(ba)bc = ca(c^{-1})bc = c(a.c)(c.b)c = c(b^{-1})(a^{-1})c = (c.b)(b.a)(a.c) = a^{-1}c^{-1}b^{-1}.$

It is clear that γ is an injective homomorphism. □

In a similar way as in Lemma 1, we can show that there does not exist any homomorphism with a finite kernel in the converse direction. Thus, we do not know whether every HCA2 can be simulated by a HCA in an elementary way or not. In the following section, we give another, non-elementary simulation.

6 Other simulations

Here, we study other, more complicated simulations. We have seen that every HCA can be simulated by a HCA2. Here we construct the converse simulation.

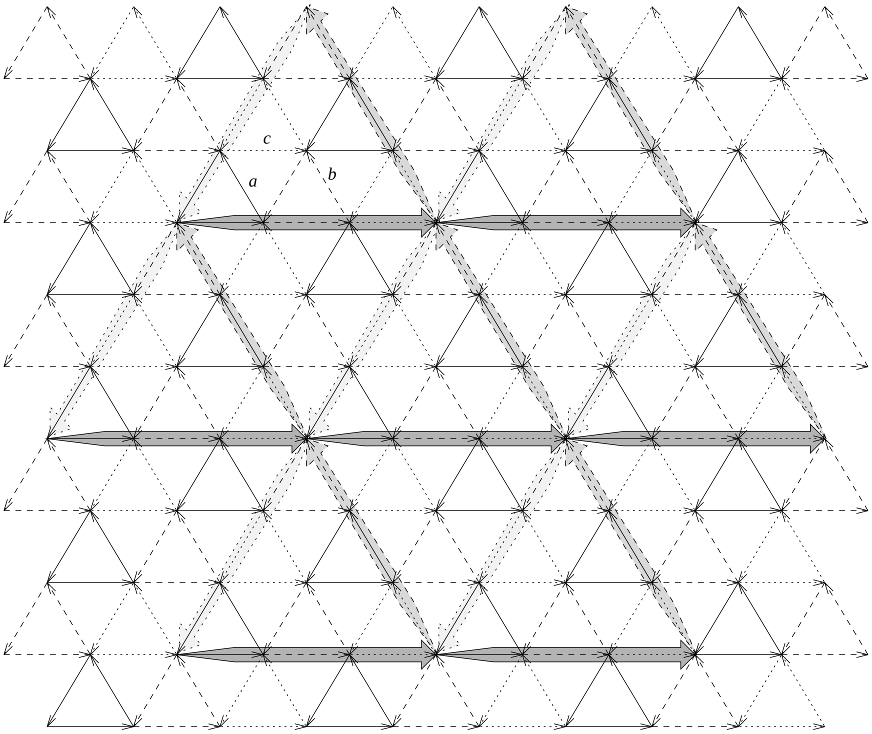


Figure 14: The injective homomorphism $\gamma : \boldsymbol{G}_h \rightarrow \boldsymbol{G}_{h_2}$.

Theorem 6 *Every HCA2 can be simulated by a HCA.*

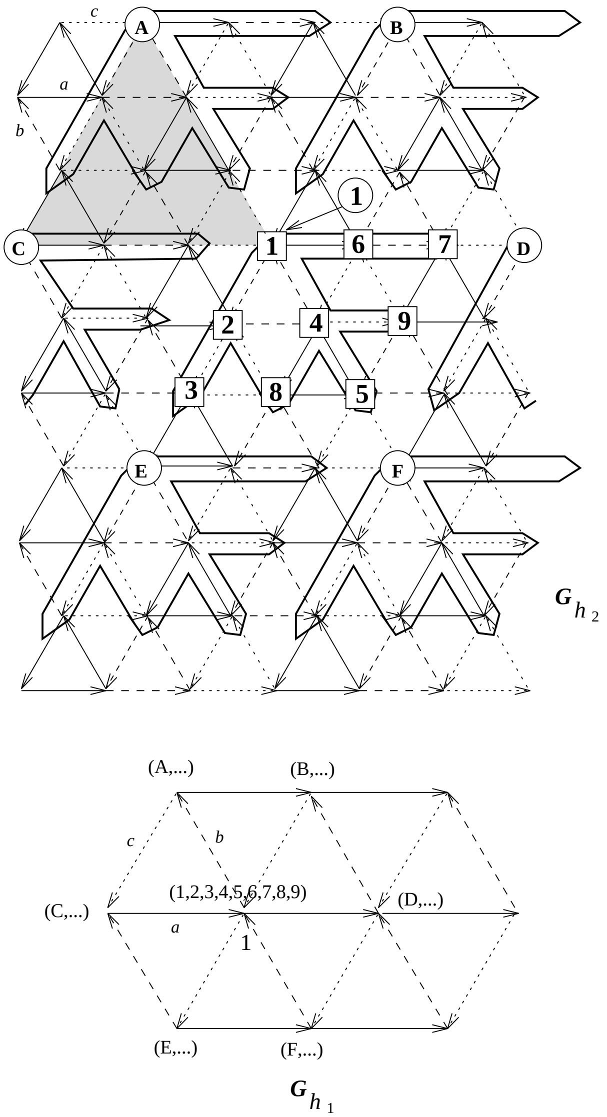
Proof. Let $\mathcal{A} = (S, \Gamma, N, \delta)$ be a HCA2 and let $c^0_{\mathcal{A}}$ be its initial configuration. Let $\gamma : \boldsymbol{G}_h \rightarrow \boldsymbol{G}_{h_2}$ be a homomorphism defined by

$$\gamma(1) = 1; \quad \gamma(a) = abc; \quad \gamma(b) = cab; \quad \gamma(c) = bca.$$

We construct a CA $\mathcal{B} = (S', \Gamma', N', \delta')$ which simulates $\mathcal{A}$: let $S = S^9$, and we define the initial configuration $c^0_{\mathcal{B}}$ of $\mathcal{B}$ by

$$c^0_{\mathcal{B}}(u) = (c^0_{\mathcal{A}}(\gamma(u)), c^0_{\mathcal{A}}(\gamma(u)b), c^0_{\mathcal{A}}(\gamma(u)bc), c^0_{\mathcal{A}}(\gamma(u)b^{-1}), c^0_{\mathcal{A}}(\gamma(u)b^{-1}a^{-1}),$$
$$c^0_{\mathcal{A}}(\gamma(u)a), c^0_{\mathcal{A}}(\gamma(u)ab), c^0_{\mathcal{A}}(\gamma(u)bc^{-1}), c^0_{\mathcal{A}}(\gamma(u)ac^{-1})).$$

The definition of this initial configuration is shown in Figure 15. The grey triangle is the image of a triangle abc of HCA, in the large shape are cells whose information are grouped in $c^0_{\mathcal{B}}$. The tuple $(1, 2, \ldots, 9)$ denotes the order of information in a state-vector of HCA. With this construction, for all information of $c^0_{\mathcal{A}}$, the neighbor information are located in a uniform way in $c^0_{\mathcal{B}}$ and at distance 1: a simulation can be given without any loss of time.



$$\gamma : G_{h_1} \xrightarrow{} G_{h_2}$$

$\gamma\,(1)=1$ $\gamma\,(a)=\ abc$ $\gamma\,(b)=\ cab$ $\gamma\,(c)=\ bca$

Figure 15: Initial configurations.

We define the transition function $\delta' : S^7 \rightarrow S$ by

$$\begin{gathered}\delta'((x_1, x_2, \ldots, x_9), (y_1, y_2, \ldots, y_9), (z_1, z_2, \ldots, z_9), (r_1, r_2, \ldots, r_9),\\ (s_1, s_2, \ldots, s_9), (t_1, t_2, \ldots, t_9), (w_1, w_2, \ldots, w_9)) =\\[1em] (\delta(w_6, w_2, y_5, t_3, w_4, r_7, w_1), \delta(r_7, w_4, w_3, r_9, w_1, w_8, w_2),\\ \delta(z_1, x_9, w_8, z_4, r_4, w_2, w_3), \delta(w_8, w_1, w_9, w_5, w_2, w_6, w_4),\\ \delta(w_4, x_3, z_7, w_8, w_9, s_1, w_5), \delta(t_3, w_7, w_4, w_1, t_8, w_9, w_6),\\ \delta(w_9, t_8, x_1, x_2, w_6, t_5, w_7), \delta(t_3, w_7, w_4, w_1, t_8, w_9, w_8),\\ \delta(x_2, w_5, w_6, w_7, x_3, w_4, w_9)).\end{gathered}$$

□

Remark that in the proof of Theorem 6, the grouped cells are

$$\{x, xb, xbc, xb^{-1}, xb^{-1}a^{-1}, xa, xab, xbc^{-1}, xac^{-1}\}$$

where x is an image element. Notice that x is in $\gamma(\boldsymbol{G}_h)$, xb is in the left-coset $\gamma(\boldsymbol{G}_h)b$ of $\gamma(\boldsymbol{G}_h)$, and so on, xac^{-1} is in $\gamma(\boldsymbol{G}_h)ac^{-1}$. In general, we can state the following theorem:

Theorem 7 *Let γ be a homomorphism from $\boldsymbol{G}$ to $\boldsymbol{G}'$. If the index of the image of $\boldsymbol{G}$ is finite, then every cellular automaton on the Cayley graph of $\boldsymbol{G}'$ can be simulated by a cellular automaton on the Cayley graph of $\boldsymbol{G}$.*

Proof. We first study the case when γ is injective. Let $\boldsymbol{H}$ be the subgroup of $\boldsymbol{G}'$ such that $\boldsymbol{H} = \gamma(\boldsymbol{G})$. Let $\mathcal{H} = \{\boldsymbol{H}, \boldsymbol{H}u_1, \boldsymbol{H}u_2, \ldots, \boldsymbol{H}u_m\}$ be the set of all distinct left-cosets of $\boldsymbol{H}$. Let $\mathcal{A} = (S, \Gamma, N, \delta)$ be a CA on the Cayley graph of $\boldsymbol{G}'$ and $c^0_{\mathcal{A}}$ its initial configuration. We define a CA $\mathcal{B} = (S', \Gamma', N', \delta')$ on the Cayley graph of $\boldsymbol{G}$ which simulates $\mathcal{A}$.

We define S' as a sup-set of S^{m+1}. In order to define the initial configuration $c^0_{\mathcal{B}}$, we group the states of $c^0_{\mathcal{A}}$ in the following way: for all v in $\boldsymbol{G}$, let

$$c^0_{\mathcal{B}}(v) = (c^0_{\mathcal{A}}(\gamma(v)), c^0_{\mathcal{A}}(\gamma(v)u_1), c^0_{\mathcal{A}}(\gamma(v)u_2), \ldots, c^0_{\mathcal{A}}(\gamma(v)u_m)).$$

We have to show, that neighbor information of $c^0_{\mathcal{A}}$ are uniformally placed in $c^0_{\mathcal{B}}$ for every components of every vectors: for all i, if for some u in $\boldsymbol{G}'$, $c^0_{\mathcal{A}}(u)$ is the i-th component in the state-vector $c^0_{\mathcal{B}}(v)$ for some v in $\boldsymbol{G}$, and a neighbor information $c^0_{\mathcal{A}}(ug)$ is the j-th component in the state-vector $c^0_{\mathcal{B}}(w)$ for some g in G' and w in $\boldsymbol{G}$ such that $w = vx$, then for all U in $\boldsymbol{G}'$ being the i-th component in a state-vector $c^0_{\mathcal{B}}(V)$ for some V in $\boldsymbol{G}$, the neighbor information $c^0_{\mathcal{A}}(Ug)$ must be the j-th component in the state-vector $c^0_{\mathcal{B}}(W)$ where W is in $\boldsymbol{G}$ and $W = Vx$.

Let $X \in \boldsymbol{G}$ and $\gamma(X) = x$. Consider cell $xu_i (\in \gamma(\boldsymbol{G})u_i)$ in $\mathcal{A}$. By the definition of $c^0_{\mathcal{B}}$, its state can be found as the i-th component in the state of cell X in $\mathcal{B}$:

$$c^0_{\mathcal{B}}(X) = (c^0_{\mathcal{A}}(x), c^0_{\mathcal{A}}(xu_1), \ldots, c^0_{\mathcal{A}}(xu_i), \ldots, c^0_{\mathcal{A}}(xu_m)).$$

In order to compute the new state of xu_i in $\mathcal{B}$, for all g in $G' \cup G'^{-1}$, the states of its neighbor cells xu_ig in $\mathcal{A}$ are needed. Let xu_ig be in $\gamma(\boldsymbol{G})u_j$ for some j. It means that there exists $\tilde{X}$ in $\boldsymbol{G}$ such that $\gamma(\tilde{X}) = \tilde{x}$ and $xu_ig = \tilde{x}u_j$, hence

$$c^0_{\mathcal{B}}(\tilde{X}) = (c^0_{\mathcal{A}}(\tilde{x}), c^0_{\mathcal{A}}(\tilde{x}u_1), \ldots, c^0_{\mathcal{A}}(\tilde{x}u_j) = c^0_{\mathcal{A}}(xu_ig), \ldots, c^0_{\mathcal{A}}(\tilde{x}u_m))$$

and $\tilde{X} = XU$ for some U in $\boldsymbol{G}$.

Let $Y \in \boldsymbol{G}$ and $\gamma(Y) = y$. Consider cell yu_i in $\mathcal{A}$. Its state can be found as the i-th component in the state of a cell Y in $\mathcal{B}$:

$$c^0_{\mathcal{B}}(Y) = (c^0_{\mathcal{A}}(y), c^0_{\mathcal{A}}(yu_1), \ldots, c^0_{\mathcal{A}}(yu_i), \ldots, c^0_{\mathcal{A}}(yu_m)).$$

We want to know, in which cell of $\mathcal{B}$ the state of the cell yu_ig of $\mathcal{A}$ can be found. Because of the properties of groups, we know that it will be the j-th component in the state-vector of some cell $\tilde{Y}$ in $\mathcal{B}$: let $\gamma(\tilde{Y}) = \tilde{y}$, then $yu_ig = \tilde{y}u_j$ and

$$c^0_{\mathcal{B}}(\tilde{Y}) = (c^0_{\mathcal{A}}(\tilde{y}), c^0_{\mathcal{A}}(\tilde{y}u_1), \ldots, c^0_{\mathcal{A}}(\tilde{y}u_j) = c^0_{\mathcal{A}}(yu_ig), \ldots, c^0_{\mathcal{A}}(\tilde{y}u_m)).$$

We must show that $\tilde{Y} = YU$.

As γ is a homomorphism, we know that

$$\gamma(\tilde{X}) = \gamma(XU) = \gamma(X)\gamma(U) = x\gamma(U) = \tilde{x}$$

and hence

$$\gamma(U) = x^{-1}\tilde{x}.$$

On the other hand, from $xu_ig = \tilde{x}u_j$, we have

$$x^{-1}\tilde{x} = u_igu_j^{-1}.$$

From $yu_ig = \tilde{y}u_j$, we have

$$\tilde{Y} = \gamma^{-1}(\tilde{y}) = \gamma^{-1}(y)\gamma^{-1}\gamma(U) = YU.$$

Let us consider the case when γ is not injective. Then there exists an injective homomorphism $\gamma' : \boldsymbol{G}/\ker(\gamma) \to \boldsymbol{G}'$, and we can construct a simulation in a similar way as before.

Remark that the simulation time factor is not always 1, it depends on the generating sets. □

Then, from Theorem 5, the following assertion holds.

Theorem 8 *Let γ be a homomorphism from a group $\boldsymbol{G}$ in another group $\boldsymbol{G}'$. If γ has a finite kernel, if the index of the image of $\boldsymbol{G}$ is finite, then every cellular automaton on the Cayley graph of $\boldsymbol{G}$ can be simulated by a cellular automaton on the Cayley graph of $\boldsymbol{G}'$ and conversely.*

Definition 7 *We shall say that $\boldsymbol{G}_1 \rhd \boldsymbol{G}_2$ if and only if there exists a homomorphism $\gamma : \boldsymbol{G}_1 \longrightarrow \boldsymbol{G}_2$ with a finite kernel such that the index of the image of $\boldsymbol{G}_1$ is finite.*

If $\boldsymbol{G}_1 \rhd \boldsymbol{G}_2$, then there exist simulations in both directions between cellular automata defined on $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$. We are interested by the symmetrized relation of $\rhd$. As simulations are transitives, we study the symmetric and transitive closure of $\rhd$.

Definition 8 *We define on finitely presented groups the relation $\boldsymbol{G}_1 \asymp \boldsymbol{G}_2$ as the symmetric and transitive closure of the relation $\rhd$.*

From Theorem 8, the following proposition holds.

Proposition 4 *$\boldsymbol{G}_1 \asymp \boldsymbol{G}_2$ if and only if there exists a suite of finitely presented groups $\boldsymbol{G}'_0, \boldsymbol{G}'_1, \ldots, \boldsymbol{G}'_n$ such that $\boldsymbol{G}'_0 = \boldsymbol{G}_1$ and $\boldsymbol{G}'_n = \boldsymbol{G}_2$ and for all $i \geq 0$, $\boldsymbol{G}'_i \rhd \boldsymbol{G}'_{i+1}$ or $\boldsymbol{G}'_{i+1} \rhd \boldsymbol{G}'_i$.*

Conjecture: If there exist a simulation between cellular automata on $\boldsymbol{G}_1$ and cellular automata on $\boldsymbol{G}_2$ in both directions, then $\boldsymbol{G}_1 \asymp \boldsymbol{G}_2$.

If this assertion is true, it would imply that the existence of simulations in both directions between cellular automata defined on two groups is an indecidable problem.

7 Cellular automata on Archimedean tilings

In the previous section we have studied hexagonal and triangular CA. Let us denote by Γ_H and Γ_T, respectively, the underlying graphs on which these CA work. T. Chaboud ([1]) has shown that Figures 16 and 17 show all possible colorings for these graphs.

The *Archimedean tilings* are presented in Figure 18; they are exactly the tilings using a finite number of regular and convex polygons such that the degree of every vertex and the order of polygons around every vertex is the same.

In [1] it is shown that all Archimedean tilings can be colored as Cayley graphs; he has also given all possible colorings.

We show now that cellular automata on all these graphs are equivalent from a computational point of view: they can be simulated by each other in a linear time. In order to show the existence of injective homomorphisms from $\boldsymbol{G}_{2\_VN}$ to groups corresponding to these tilings, we introduce the following notion.

Definition 9 *Let Γ be an Archimedean tiling colored as a Cayley graph. Consider vertex 1, a generator g. Let us denote by $\vec{x}$ the arc starting at 1 colored by g and considered as a vector in $\mathbb{R}^2$. Let A be a vertex and h a generator or an inverse generator. We denote by $\widehat{(\vec{x}, h)}_A$ the angle between $\vec{x}$ and the arc colored*

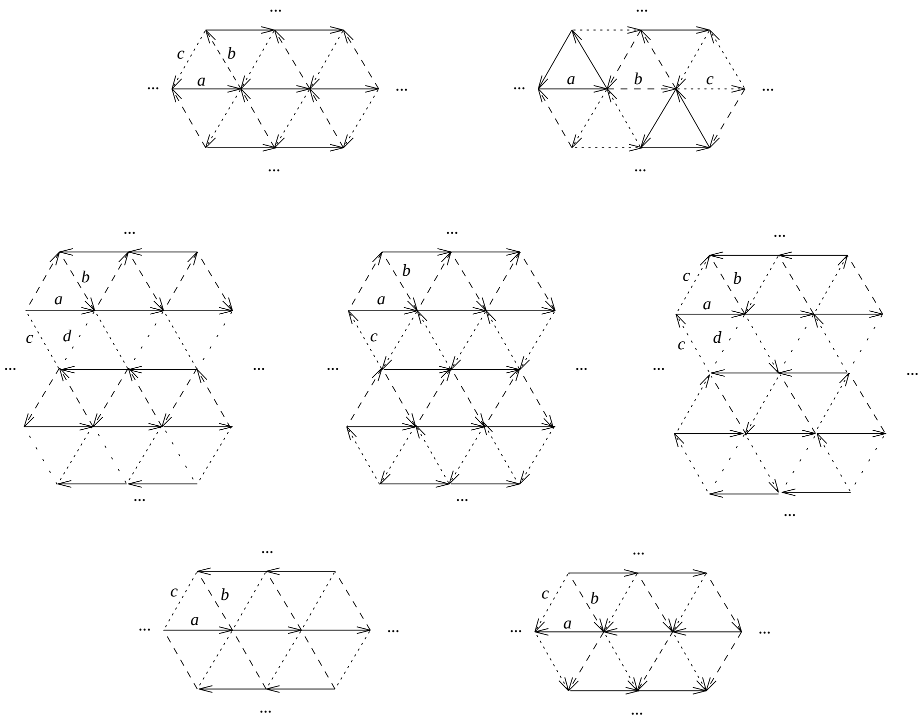


Figure 16: All Cayley graphs for hexagonal CA.

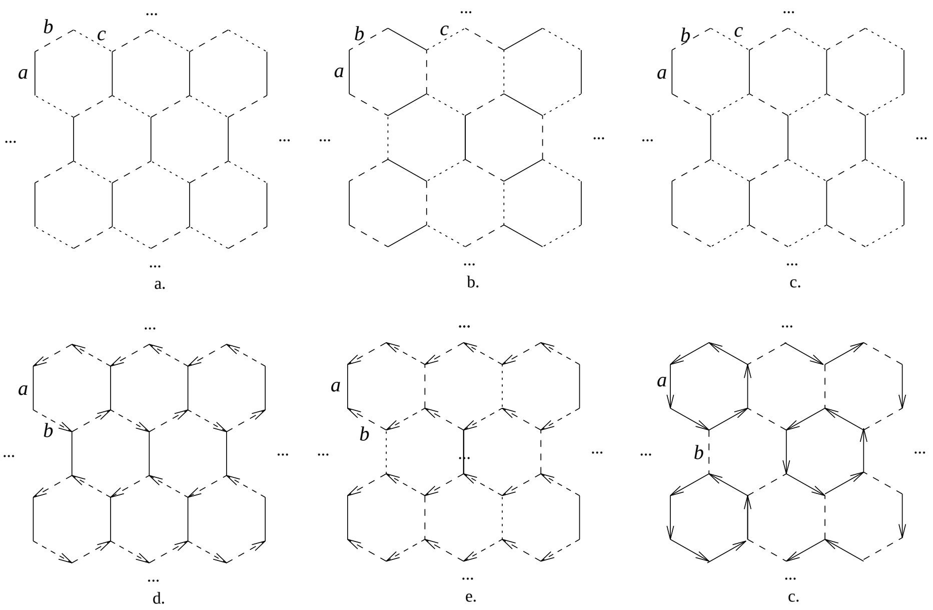


Figure 17: All Cayley graphs for triangular CA.

by h starting at A. We say that two vertices A and B have the same situation *if for all generator and inverse generator h, $\widehat{(\vec{x},h)}_A = \widehat{(\vec{x},h)}_B$.*

See an example for same and different situations in Figure 19. Vertices A and B have the same situation, but not A and C.

The following lemma is a consequence of the facts that in an Archimedean tiling, the type of vertices is the same ([4]) and that the geometrical order of generators in every vertex is the same ([1]).

Lemma 4 *Let x and y be two vertices having the same situation. Then for all generator (or inverse generator) g, xg and yg have the same situation.*

Now we can show the following proposition:

Proposition 5 *There is an injective homomorphism from $\boldsymbol{G}_{2\_VN}$ to all groups whose Cayley graphs have an underlying graph corresponding to an Archimedean tiling.*

Proof. Let Γ be the Cayley graph of a group $\boldsymbol{G}$ such that its underlying graph corresponds to an Archimedean tiling. Let x, y be two vertices having the same situation and let p_1 be a path from x to y: $y = xp_1$. Let z be a third vertex having the same situation as x and y and let p_2 be a path from x to z: $z = xp_2$.

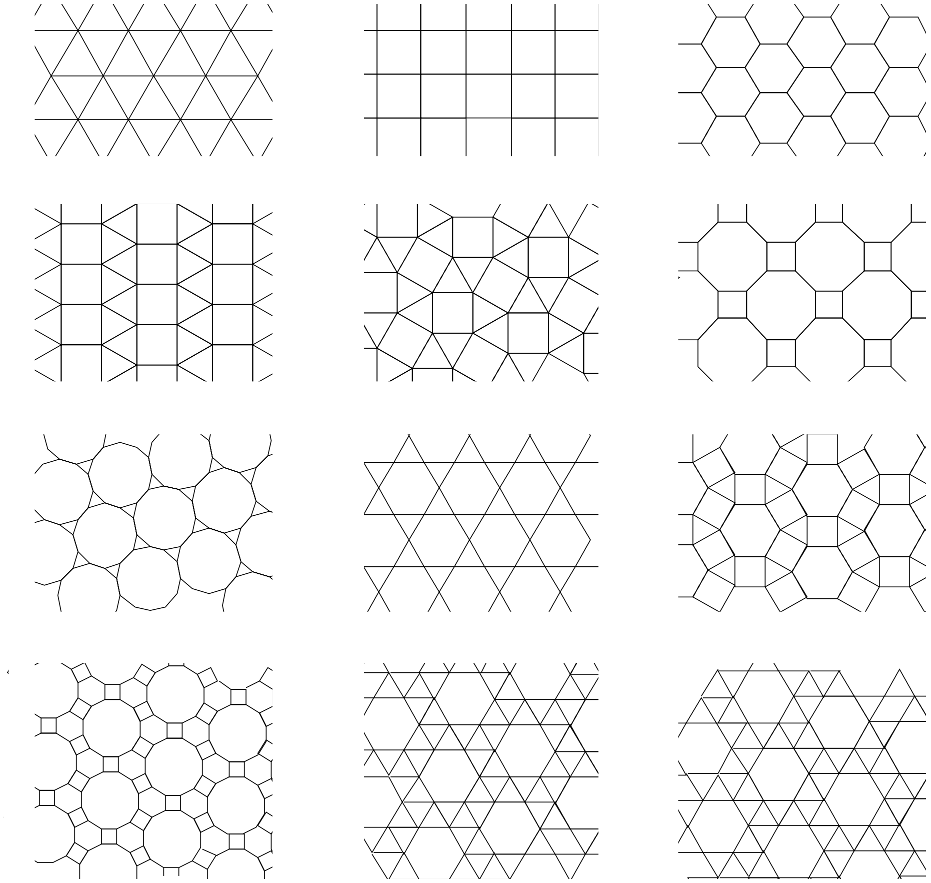


Figure 18: Archimedean tilings.

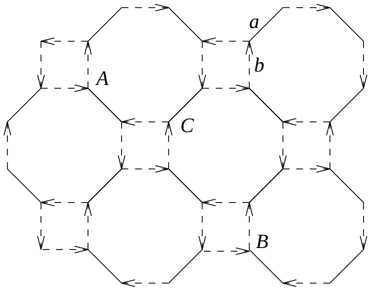


Figure 19: Different vertex-situations.

We suppose that x, y and z are chosen in such a way that $p_1^n \neq p_2^{n'}$ for all $n > 0$ and $n' > 0$ in $\mathbb{N}$. It is possible, because these tilings are periodic in the plane in two independant directions. Then, from Lemma 4, for all $n > 0$ in $\mathbb{N}$, $p_1^n \neq 1$ and $p_2^n \neq 1$. On the other hand, also from Lemma 4, the path p_1 starting at z is "parallel" to the path between x and y, and the path p_2 starting at y is "parallel" to the path between x and z, hence, $zp_1 = yp_2$, that is, $p_1p_2 = p_2p_1$.

Recall that two-dimensional von Neumann CA are defined on the Cayley graph of the group $\boldsymbol{G}_{2\_VN} = \langle a, b \mid ab = ba \rangle$. Let $\gamma : \boldsymbol{G}_{2\_VN} \rightarrow \boldsymbol{G}$ be a mapping defined by

$$\gamma(1) = 1, \quad \gamma(a) = p_1, \quad \gamma(b) = p_2, \quad \text{and}$$
$$\gamma(uv) = \gamma(u)\gamma(v) \quad u, v \in \boldsymbol{G}_{2\_VN}.$$

It is clear that γ is an injective homomorphism. □

Then, from Theorems 5 and 7 and Proposition 5, the following assertion holds.

Theorem 9 *Every two-dimensional von Neumann cellular automaton can be simulated by a cellular automaton on any Archimedean tiling and conversely.*

This result can also be interpreted in the following way. If we consider Cayley graphs as possible architectures for parallel machines, we can choose any of Archimedean tilings for such an architecture in the plane, they have the same computational power. However, as simulations between cellular automata on these graphs require many states, it is necessary that machines have a sufficient amount of local memory.

8 Open problems

In this paper, we have only studied simulations, where the states of the simulated cellular automaton are considered as atomic informations. However, many simulations exist with splitting states, they should also be studied.

We have given a sufficient condition for converse simulations between cellular automata on Cayley graph. Can any sufficient and necessary condition be given? This is the question that we have asked ourselves when we have defined the relation $\asymp$ between groups. If the answer is "no" in the general case, whether does a subclass of groups exist for which such a condition can be given?

References

[1] T. Chaboud. Planar Cayley graphs. *preprint*, 1994.

[2] S. Cole. Real time computations by n-dimensional iterative arays of finite-state machines. *IEEE Transactins on Computers*, 4:349–365, 1969.

[3] U. Frisch, D. d'Humières, B. Hasslacher, P. Lallemand, Y. Pomeau, and J.-P. Rivet. Lattice gas hydrodynamics in two and three dimensions. *Complex Systems*, 1(4):649–708, 1987.

[4] B. Grünbaum and G. C. Shephard. *Tilings and Patterns*. Freeman and Co, 1987.

[5] A. Machí and F. Mignosi. Garden of Eden configurations for cellular automata on Cayley graphs of groups. *SIAM Journal on Discrete Mathematics*, 6(1):44–56, 1993.

[6] Zs. Róka. One-way cellular automata on Cayley graphs (short version). In *FCT'93*, volume 710 of *LNCS*, pages 406–417. Springer Verlag, August 1993.

[7] Zs. Róka. *Cellular automata on Cayley graphs*. PhD thesis, Ecole Normale Supérieure de Lyon, France, July 1994.

[8] Zs. Róka. One-way cellular automata on Cayley graphs. *Theoretical Computer Science*, 132:259–290, 1994.