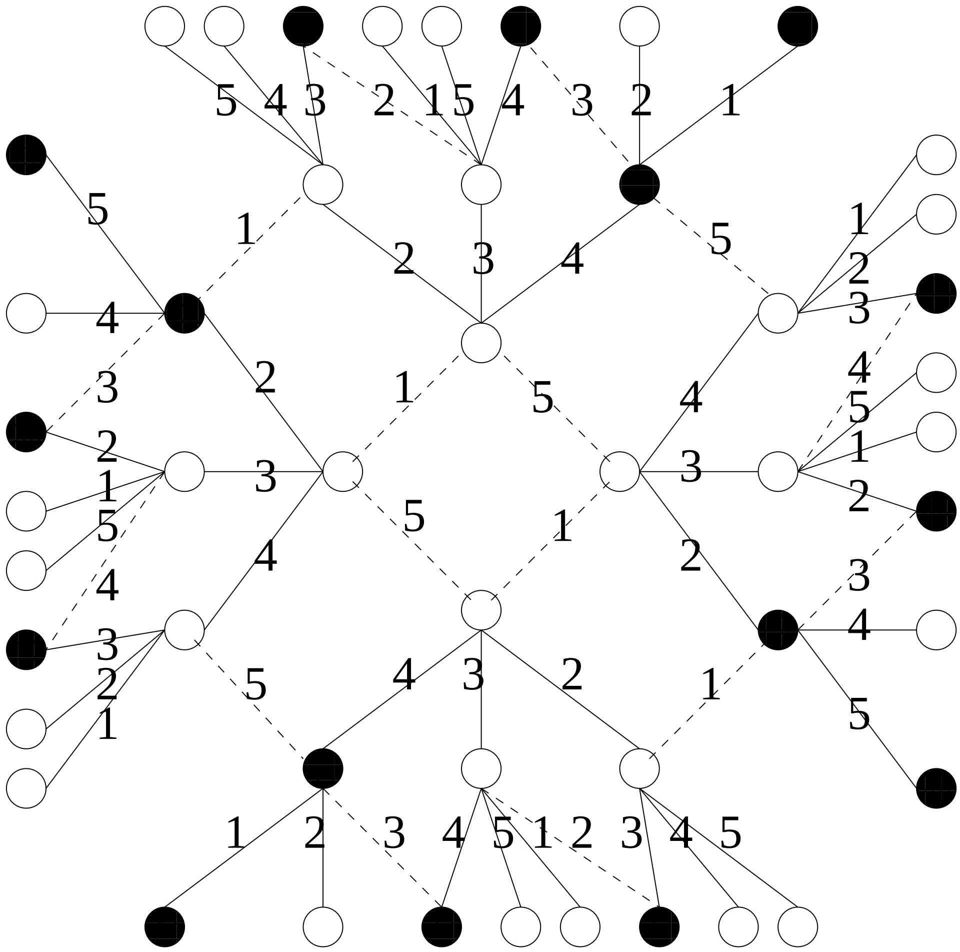

Cellular automata in the hyperbolic plane.

Maurice Margenstern

Publications du GIFM, N° 99-103

Service de reprographie de l'I.U.T. de Metz

ISBN 2-9511539-3-7 **IUT**

Groupe d'Informatique Fondamentale de Metz I.U.T. de Metz
Directeur de la publication : Maurice Margenstern

Cellular automata in the hyperbolic plane.

Maurice Margenstern

G.I.F.M., Université de Metz,
I.U.T. de Metz, Département d'Informatique,
Île du Saulcy,
57045 Metz Cedex, France,
e-mail : margens@iut.univ-metz.fr

Abstract

In this paper, we define a new way for locating the cells of a cellular automaton on the grid of the hyperbolic plane, which is based on the tessellation obtained from the regular pentagon with right angles: the pentagrid. We provide the basic algorithms from which programmig tools can be easily derived in order to devise concrete cellular automata on the pentagrid.

1. Introduction

Outside a paper by late Robinson, it seems that a very few number of papers in theoretical computer science were devoted to hyperbolic spaces.

The first exception that we know, is a recent paper by the present author and Kenichi Morita.

The goal of the present paper is to draw the attention of the researchers that deal with cellular automata on the possible interest to investigate the behaviour of cellular automata in that rather new environment which has very different properties than the euclidian one, especially in two dimensions and also in higher ones. The main goal of the paper is to provide tools to devise cellular automata in that new setting where our usual intuition is of no use, especially when it is based on the euclidean orientation.

One could point at the abstractness of the consideration of cellular automata in the hyperbolic plane as a too much theoretical point of view. It can be replied to such an objection in both directions. The first one is that the investigations of *linear* cellular automata, see for instance the works of the Bremen school and of the Milano school, [1, 3], has lead to very interesting theoretical works which, in particular, gave an explanation of experimental results, coming from researches that where inspired by considerations belonging to physics. Another direction is that hyperbolic geometry is a *natural* geometry, which is currently used in cosmology, even if our surrounding euclidean geometry leads us to believe that it is the single possible *natural* geometry. A third argument is that, as it is shown in the paper of the present author with Kenichi Morita, the hyperbolic plane provides a *natural* setting in which it is possible to solve NP-complete problems in polynomial time.

The tool box given in the paper deals with easy properties of elementary arithmetics which are recalled in order to give a self content presentation. In our conclusion, we shall indicate some features which concern higher dimensions.

2. Cellular automata in the hyperbolic plane

In this section, we recall the definition and some properties of the hyperbolic plane. In particular, we define the Poincaré disk model, which, historically, played an important part in the development of the mathematical studies on hyperbolic geometry.

Then we draw the attention of the *regular tilings* of the hyperbolic plane.

In the euclidean plane, there are three such tilings: they are generated, respectively, by te equilateral triangle, the square and the regular hexagon. In contrary to that situation, the hyperbolic plane can be divided by regular tilings in infinitely many ways. There are already infinitely many *triangular* regular tilings. Starting from number 5, any regular polygon with n sides and with right angles generates a regular tiling of the hyperbolic plane. Other regular tilings are possible: as an example, besides the regular tiling generated by the regular pentagon with right angles, there is another one, generated by the regular pentagon with angle $2\pi/5$.

As a first frame for investigations, we suggest to focus the attention on the simplest case, the case of the regular pentagon with right angles. To some extent, that case is the transposition of the square into the hyperbolic plane. It is not possible to display three regular polygons around a point as it is in the euclidean plane with the regular hexagon. It is possible to do that with at least four regular polygones. For four regular polygons, there are infinitely many solutions: each one is generated by the regular polygon with right angles. The simplest one is the pentagon.

We refer the reader to [8] for a short introduction to the hyperbolic plane. For more details, we refer the reader to the literature pointed in [8], for instance to [10] and [12].

2.1 The hyperbolic plane: Poincaré's disk

Lines of the hyperbolic plane and angles

In Poincaré's disk model, the hyperbolic plane is the set of points lying in the open unit disk of the euclidean plane. The lines of the hyperbolic plane in Poincaré's disk model are either the intersection of diametral segments with the unit disk (points lying on the unit circle do not belong to the hyperbolic plane) or the intersections with it of circles that are orthogonal to the unit circle. We say that the considered circle *supports* the hyperbolic line, h-line for short, and sometimes simply *line* when there is no ambiguity.

Consider the points of the unit circle as *points at infinity* for the hyperbolic plane: it is easy to see that an h-line defines two points at infinity by the intersection of its euclidean support with the unit circle. They are called points at infinity of the h-line. The following easily proved properties will often be used: any h-line has exactly two points at infinity; two points at infinity define a unique h-line passing through them; a point at infinity and a point in the hyperbolic plane uniquely define an h-line.

The angles between h-lines are defined as the euclidean angle between the tangents to the arcs which are taken as the support of the corresponding h-lines. This is one reason for choosing that model: hyperbolic angles between h-lines are, in a natural way, the euclidean angle between the corresponding supports. In particular, orthogonal circles support perpendicular h-lines.

In the hyperbolic plane, given a line, say ℓ, and a point A not lying on ℓ, there are infinitely many lines passing through A which do not intersect ℓ. In the euclidean plane, two lines are parallel if and only if they do not intersect. If the points at infinity are added to the euclidean plane, parallel lines are characterized as the lines passing through the same point at infinity. Hence, as for lines, to have a common point at infinity and not to intersect is the same property in the euclidean plane. This is not the case in the hyperbolic plane, where two lines may not intersect and have no common point at infinity. We shall distinguish those two cases by calling *parallel*, h-lines that share a common point at infinity, and *non secant*, h-lines which have no common point at all neither in the hyperbolic plane nor at infinity. So, considering the situation illustrated by figure 1 below, there are exactly two h-lines parallel to a given h-line that pass through a point not lying on the latter line and infinitely many ones that pass through the point but are non-secant with the given h-line. This is easily checked in Poincaré's disk model, see figure 1. Some authors call *hyperparallel* or *ultraparallel* lines that we call *non-secant*.

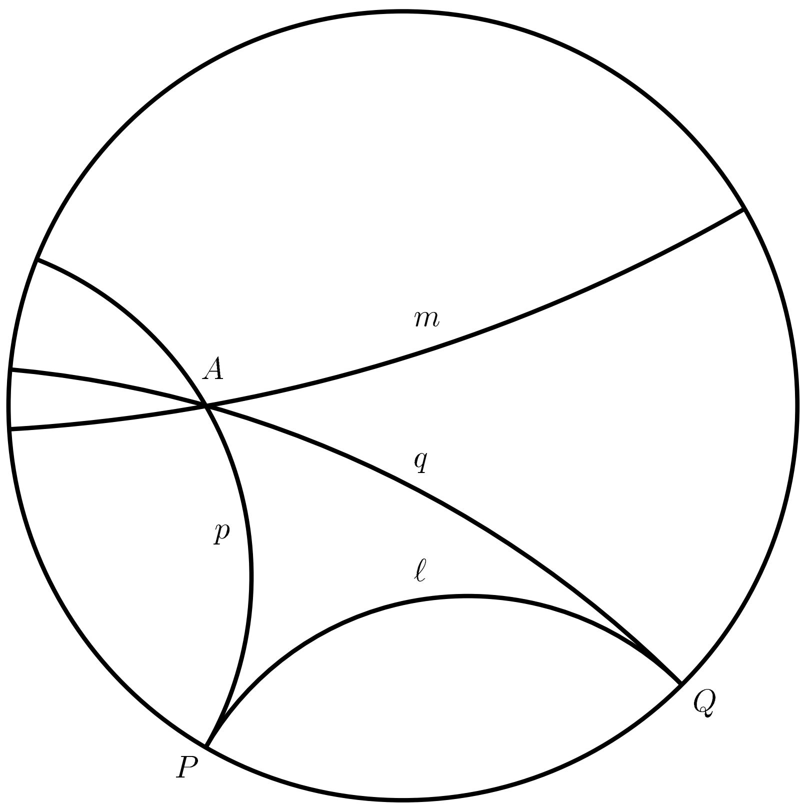

Figure 1 Lines p and q are parallel to line ℓ, with points at infinity P and Q. h-line m is non-secant with ℓ.

Another aspect of the parallel axiom lies in the sum of interior angles at the vertices of a polygon. In the euclidean plane, the sum of angles of any triangle is exactly π. In the hyperbolic plane, this is no more true: the sum of the angles of a triangle is *always less* than π. The difference with π is, by definition, the *area* of the triangle in the hyperbolic plane. Indeed, one can see that the difference of the sum of the angles of a triangle with π has the additive property of a measure on the set of all triangles. As a consequence, there is no rectangle in the hyperbolic plane. Consequently two non-secant lines, say ℓ and m, have, at most, one common perpendicular. It can be proved that this is the case: two non-secant lines of the hyperbolic plane have exactly one common perpendicular.

As that result plays an important part in the proof of the existence of the grid which we shall later define in the hyperbolic plane, we give a sketchy proof of that fact here.

The proof is given in the frame of Poincaré's model, although there are proofs purely belonging to hyperbolic geometry, see for instance [10]. The notion of *pencils of h-lines* is very important in hyperbolic geometry. Such a pencil is a family of h-lines that depend on one parameter. There are three kinds of them, in the hyperbolic plane: the set of lines that pass through the same point of the hyperbolic plane, the set of lines that pass through the same point at infinity $-$ they are indeed parallel $-$ and the set of lines orthogonal to a given line then called *basis* of the pencil.

That latter family is characterized by the set of the centers of the euclidean supports of the lines which belong to the pencil. It is the set of centers of circles orthogonal to two given circles, which are here the unit circle and the support of the h-line that defines the basis of the pencil. In our case, these circles intersect and the searched set is the euclidean line that pass through the two common points of these circles, except the open segment inside the both circles. In elementary geometry, this is called the *radical axis* of the two circles. If two h-lines, say ℓ and m do not intersect, the radical axes they define with the unit circle do intersect in a point P, possibly at infinity. If P is at finite distance, the tangents from P to the unit circle touch it in R and S. The intersection with the unit open disk of the circle centered in P that passes through R and S gives us the needed common perpendicular. If P is at infinity, the circles are symmetric to the diameter of the unit circle which passes through P, say Δ, and the just given construction gives the diameter, which is perpendicular to Δ, as the common perpendicular of ℓ and m.

It can be added that parallel h-lines have no common perpendicular.

Consider two parallel h-lines, and let P be their common point at infinity. The above construction would give P as the center of the circle supporting the common perpendicular. But no circle with center P is orthogonal to the unit circle, except the circle with zero as a radius.

Consider the following problem of euclidean geometry:

(P) Let α, β, γ be positive real numbers such that $\alpha+\beta+\gamma = \pi$ and let be given two lines ℓ, m intersecting in A with angle α. How many triangles ABC can be constructed with $B \in \ell$, $C \in m$ and BC making angle β in B with ℓ?

The answer is clearly: infinitely many. That property of the euclidean plane defines the notion of *similarity.*

Another consequence of the non-validity of Euclid's axiom on parallels in the hyperbolic plane is that there is no notion of similarity in that plane: *if α, β, γ are positive real numbers such that $\alpha+\beta+\gamma < \pi$, ℓ and m are h-lines intersecting in A with angle α, there are exactly two triangles ABC such that $B \in \ell$, $C \in m$ and BC makes angle β in B with ℓ and angle γ in C with m.*

Each of those triangles is determined by the side of ℓ with respect to A in which B is placed.

Reflections with respect to a h-line

Any h-line, say ℓ, defines a *reflection* denoted ρ_ℓ through that line. Let Ω be the center of the euclidean support of ℓ, R its radius. Two points M and M' are *symmetric* with respect to ℓ if and only if Ω, M and M' belong to the same euclidean line and if $\Omega M.\Omega M' = R^2$. Moreover, M and M' do not lie in the same connected component of the complement of ℓ in the unit disk. We also say that M' is obtained from M by the reflection through ℓ. It is clear that M is obtained from M' by the same reflection.

These definitions allow us to prove the property of the existence of a common perpendicular for non-secant lines.

Indeed: let a line that passes through P intersect ℓ in A_ℓ and B_ℓ. As P belongs to the radical axis, $PA_\ell.PB_\ell = PR^2$. The same relation holds for a line which passes through P and intersect m in A_m and B_m. This entails that B_m and B_ℓ are the reflection of, respectively A_m and A_ℓ through the circle σ centered in P and passing through R and S, which is obviously orthogonal to the unit circle. From what we said, the image of ℓ through ρ_σ is ℓ and the same is true for m. Consequently, ℓ and m are both perpendicular to σ.

All the transformations of the hyperbolic plane that we shall later consider are reflections or constructed by reflections.

By definition, an *isometry* of the hyperbolic plane is a finite product of reflections. Two segments AB and CD are called *equal* if and only if there is an isometry transforming AB into CD.

It is proved that finite products of reflections can be characterized as either a single reflection or the product of two reflections or the product of three reflections. In our sequel, we will mainly be interested by single reflections or products of two reflections (the reader interested in the properties of a product of three reflections is referred to the literature).

At this point, we can compare *reflections* through a line in the hyperbolic plane with symmetries with respect to a line in the euclidean plane. Indeed, these respective transformations share many properties on the objects on which they respectively operate. However, there is a very deep difference between the isometries of the euclidean plane and those of the hyperbolic plane: while in the first case, the group of isometries is solvable, in the second case, it is simple.

The product of two reflections with respect to lines ℓ and m is a way to focus on that difference. In the euclidean case, according to whether ℓ and m

do intersect or are parallel, the product of the two corresponding symmetries is a rotation around the point of intersection of ℓ and m, or a shift in the direction perpendicular to both ℓ and m. In the hyperbolic case, if h-lines ℓ and m intersect in a point A, the product of the corresponding reflections is again called a rotation around A as far as the obtained transformation can be considered as what is intuitively called a rotation. But, if ℓ and m do not intersect, there are two cases: either ℓ and m intersect at infinity, or they do not intersect at all. This gives rise to different cases of shifts. The first one, called *shift*, is a kind of degenerated rotation, as in the euclidean case, and the second one is called *hyperbolic shift* or *displacement* along the common perpendicular to ℓ and m. Such a displacement can be characterized by the image P' of any point P on the common perpendicular, say n. We shall speak of the displacement along n transforming P into P' and sometimes simply of *hyperbolic shift* or *displacement* when the explicit indication of the line along which the displacement is performed is not needed.

It can be proved that for any couple of two h-lines ℓ and m, there is an h-line n such that ℓ and m are exchanged in the reflection through n. In the case when ℓ and m are non-secant, n is the perpendicular bisector of the segment that joins the intersections of ℓ and m with their common perpendicular. The construction of h-line n is the same in all cases: let L_1, L_2 be the points at infinity of ℓ and M_1, M_2 those of m. We can always label the points in such a way that the euclidean lines L_1M_1 and L_2M_2 intersect in Ω, outside the closed unit disk. In the case that $L_1 = M_1$, the euclidean line L_1M_1 is the tangent at L_1 to the unit circle. From Ω, the tangents to the unit circle define two points at infinity through which a single h-line passes which is namely h-line n. Indeed, it is not difficult to check that L_1 and L_2 are the images of, respectively, M_1 and M_2 under the reflection defined by n constructed as just indicated. It immediately ensues that ℓ is the image of m under the reflection through n.

2.2 The pentagrid

Definition and existence of the pentagrid

Our pentagrid is obtained by *tessellation*. That operation first consists in taking a convex polygon P. Let $\mathcal{S}(P)$ be the set of the h-lines that support its sides. If $\mathcal{E}$ is a set of polygons, one extends $\mathcal{S}$ to $\mathcal{E}$ by setting that $\mathcal{S}(\mathcal{E}) = \bigcup\limits_{P\in\mathcal{E}} \mathcal{S}(P)$. Given $\mathcal{K}$ a set of h-lines and $\mathcal{E}$ a set of polygons, we define that $\rho_{\mathcal{K}}(\mathcal{E}) = \underset{k\in\mathcal{K},Q\in\mathcal{E}}{\cup} \rho_k(Q)$. Setting $\mathcal{T}_0 = \{P\}$, we inductively define $\mathcal{T}_{k+1}$ by $\mathcal{T}_{k+1} = \rho_{\mathcal{S}(\mathcal{T}_k)}(\mathcal{T}_k)$. Finally, we define $\mathcal{T}^* = \overset{\infty}{\underset{k=0}{\cup}} \mathcal{T}_k$ to be the *tessellation* generated by P. We say that the tessellation is a *tiling* one if and only if the following conditions hold:

- any point of the hyperbolic plane belongs to at least one polygon in $\mathcal{T}^*$;
- the interiors of the elements of $\mathcal{T}^*$ are pairwise disjoint.

In the so defined tiling of the hyperbolic plane, all tiles are equal to the generating polygon.

It was proved by Poincaré, [11], that any triangle with angles π/ℓ, π/m, π/n with $\dfrac{1}{\ell} + \dfrac{1}{m} + \dfrac{1}{n} < 1$ generates a unique tiling tessellation. There are elementary proofs of the result in [2] and [9]. The theorem immediately shows that a tiling tessellation is generated by the regular pentagon with right angles which is generated by the triangle with the following angles : $\dfrac{\pi}{5}, \dfrac{\pi}{4}, \dfrac{\pi}{2}$. One can easily check that this pentagon, if it exists, can be split into ten of such triangles. Conversely, as the angle condition is satisfied, the regular pentagon with right angles can be built up from the considered triangles. This also shows that the regular pentagon with right angles exists and is unique, up to isometries. The so defined tiling tessellation will from now on be called *pentagrid*.

The Fibonacci tree

In [8], we give the material for an independent proof of the existence of the pentagrid. That material is based on a bijection which is estabished between the tiling of the south-west quarter of the hyperbolic plane with a special infinite tree: the *Fibonaci tree*.

Here, we recall the construction of the tree associated to a part of the pentagrid. The proof that the construction is in bijection on a part of the pentagrid is given in [8] and the reader is referred to it for further details.

Consider figure 2, below, where four regular pentagons with right angles have a common vertex A and, pair-wise, a common side ending with A. It is plain that, by the reflections through x- and y-axes, and by starting from P_0, we obtain the three other pentagons. And so, the tessellation exists if and only if it exists for a quarter of the hyperbolic plane.

Let us now restrict ourselves to such a quarter, say the south-western one, for instance, and denote it by $\mathcal{Q}$. Figure 3 shows how the pentagrid looks for a quarter of the hyperbolic plane.

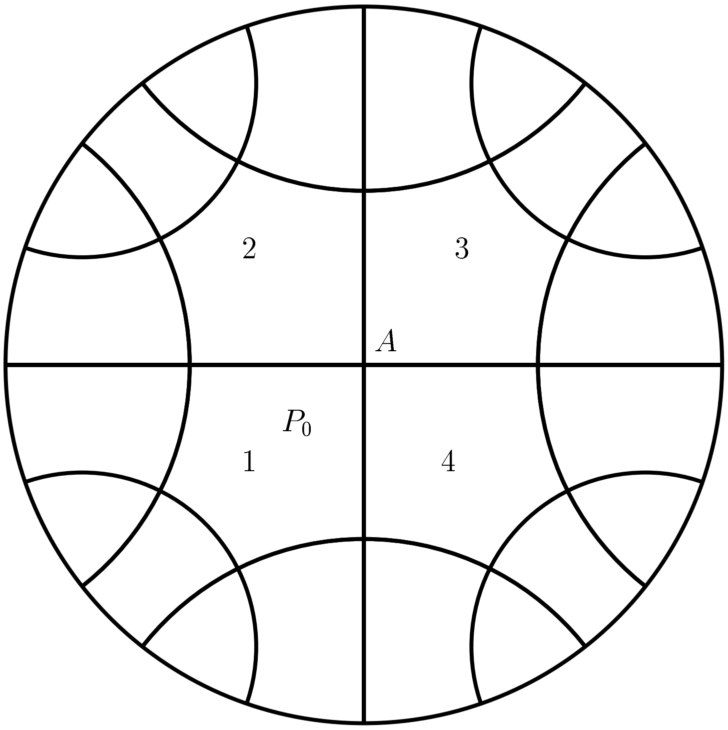

Figure 2 The four roots of the pentagrid

Consider again P_0 in $\mathcal{Q}$ − call it the *leading* pentagon of $\mathcal{Q}$ − and number its sides clockwise by **1**, **2**, **3**, **4** and **5** as indicated on figure 4. By the existence of a common perpendicular between two edges of the pentagon which are separated by a single edge, we see that h-lines **2** and **3** do not intersect **5**, which means that one component in the unit disk, that is delimited by the complement of **3**, is included in $\mathcal{Q}$, namely the component which does

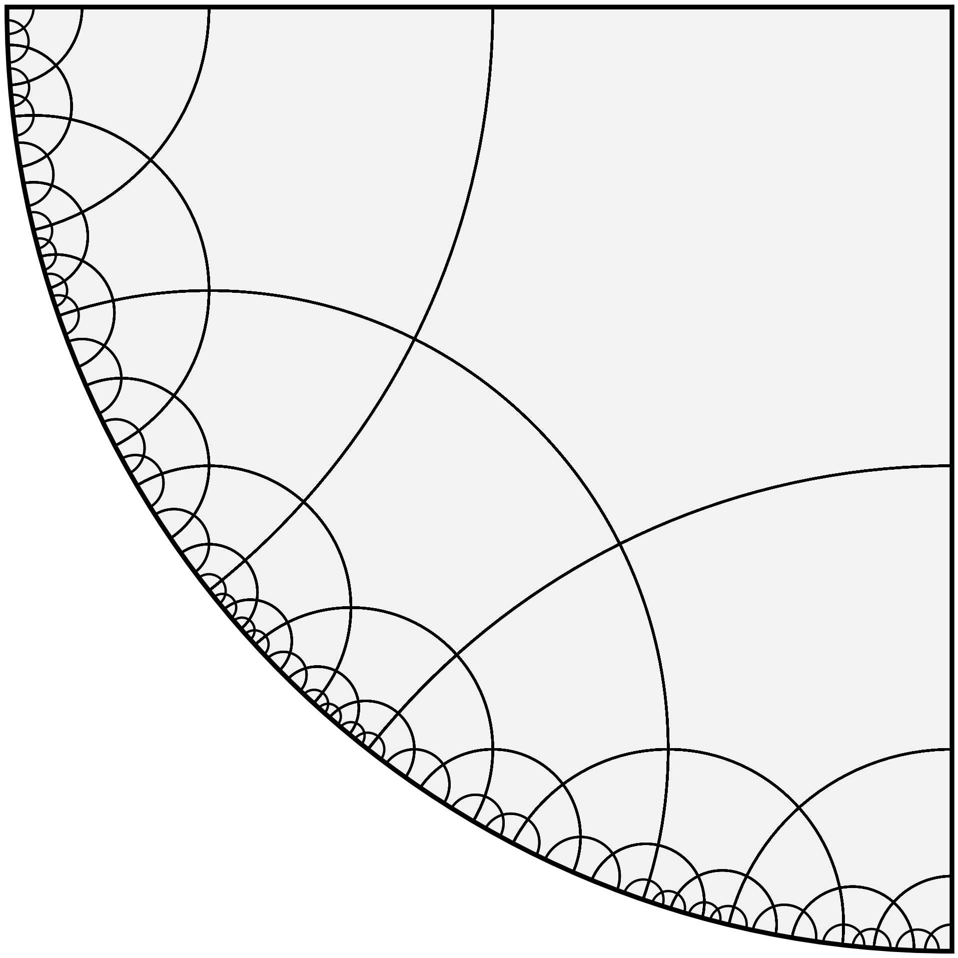

Figure 3 The pentagrid in the south-western quarter

not intersect P_0. The complement of P_0 in the quarter can be split into three regions as follows. The support of **2** delimits two connected components in the quarter. Take as R_1 the closure of the component which does not intersect P_0. What remains in $\mathcal{Q}\backslash R_1$ is in the same way separated by **3** and, similarly, call R_2 the region delimited by **3** whose interior does not intersect P_0. Define R_3 in the same way with respect to **4**, R_1 and R_2 being deleted from $\mathcal{Q}$, see figure 4.

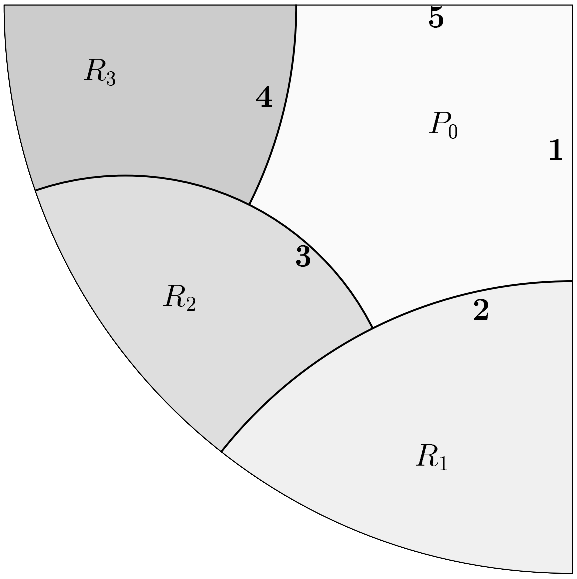

Figure 4 Splitting the quarter into four parts

One can easily notice that R_1 and R_2 are the images of $\mathcal{Q}$ by, respectively, the displacement along **1** that transforms **5** into **2** and the displacement along **4** that transforms **5** into **3**. In order to see these transformations most conveniently, take note of the fact that **1** is the common perpendicular of **5** and **2**, and that **4** is the common perpendicular of **5** and **3**. This is an immediate consequence of the fact that P_0 has the right angle as an interior angle for all its vertices. This defines the initial part of a tree: P_0 is associated to the root of the tree, and let us consider that the root has three sons, ordered from the left to the right and respectively associated to R_3, R_2 and R_1. We can denote it as indicated by figure 6. We shall say that the root is a 3-node because it has three sons.

Now, observe that regions R_1 and R_2 are isometric to $\mathcal{Q}$ by the displacements described above. This means that the same splitting can be done for these regions, which means that sons can be defined for the corresponding nodes in the same way as it was performed for the root. We shall also say that the nodes associated to R_1 and R_2 are also associated to their leading pentagons which are the images of P_0 by the considered displacements.

Now, let us see the status of region R_3. It is plain that R_3 *is not* isometric to $\mathcal{Q}$. But let us consider P_1 the reflection of P_0 through **4** with sides which are now numbered anticlockwise, so that the same number is given to the edges supported by the same h-line. In order to avoid possible confusion, we put the name of the considered pentagon as an index, if needed. Define P_1 as the leading pentagon of R_3. We notice that $\mathbf{2}_{P_1}$ delimits the image of $\mathcal{Q}$ by the displacement along $\mathbf{1}_{P_1}$ that transforms $\mathbf{5}$ into $\mathbf{2}_{P_1}$. Let S_1 be the considered region. Now $\mathbf{1}_{P_1}$ delimits S_2 which is the closure of $R_3 \backslash (S_2 \cup P_1)$. And so, we can associate two sons to the node associated to R_3 : the left one to S_2 and the right one to S_1. Each thus defined node is also associated to the leading pentagon of the corresponding region. We say that the node associated to R_3 is a 2-node. Take note of the fact that $S_2 \cup P_1$ is obtained from $R_3 \cup P_0$ by the displacement along $\mathbf{5}_{P_0}$ that transforms $\mathbf{1}_{P_0}$ into $\mathbf{4}_{P_0}$. See figure 5.

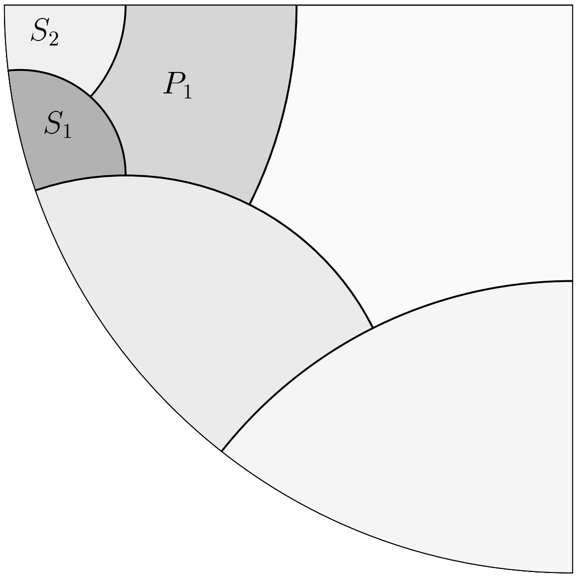

Figure 5 Splitting region R_3 into three parts

One can clearly see how me may proceed now. Define the following two rules:

- a 3-node has three sons: on the left a 2-node and, in the middle and on the right, in both cases, 3-nodes;
- a 2-node has 2 sons: the left one is again a 2-node and the right one a 3-node.

Those two rules, combined with the axiom which tells that the root is a 3-node, uniquely define a tree which we call *Fibonacci* tree.

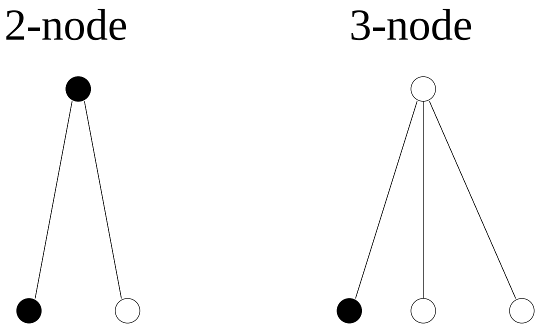

Figure 6 Two rules for the Fibonacci tree

Now, define by s_n the number of nodes of depth n in a Fibonacci tree, the root being of depth 0, as usual. Define d_n, t_n as, respectively, the numbers of 2- and 3-nodes of depth n. It is plain that $s_n = d_n + t_n$. By the very definitions, it is easy to see that $d_{n+1} = d_n + t_n = s_n$ and $t_{n+1} = d_n + 2.t_n = t_n + s_n$. And so, if we assume that $d_n = F_{2n-1}$, $t_n = F_{2n}$ and $s_n = F_{2n+1}$, where F_n is the Fibonacci sequence, we easily obtain that $d_{n+1} = F_{2n+1}$, $t_{n+1} = F_{2n+2}$ and $s_{n+1} = F_{2n+3}$. As the relation holds for $n = 0$, we proved, by induction on n that the number of nodes of the same level n is F_{2n+1} for all n. Whence the name of Fibonacci tree.

In order to set up a bijection with the considered tiling of $\mathcal{Q}$, one may simply label the edges of the tree, which is considered as a non-directed graph, in the following way: the label, leading from a node to one of its sons, is the number in $\{\mathbf{1},\mathbf{2},\mathbf{3},\mathbf{4},\mathbf{5}\}$ of the reflection that transforms its associated pentagon into the pentagon associated to the considered son.

From the numbering of the first three rows in figure 7, we can see the following two properties, which we shall further prove by induction :

- odd rows start with 4 and are decreasing ;
- even rows start with 1 and are increasing.

Call *missing numbers* the numbers which are indicated over dotted edges in figure 7. These arcs designate a reflection that directly transforms the pentagon associated to the above node into the pentagon associated to the below node, but that corresponds to an arc which does not belong to the tree. We can also easily see that if we mix the numbering of a row with the missing numbers as indicated above and at the corresponding place, we obtain a regular sequence of numbers : a beginning of $(43215)^{+\infty}$ for odd rows and a beginning of $(12345)^{+\infty}$ for even rows.

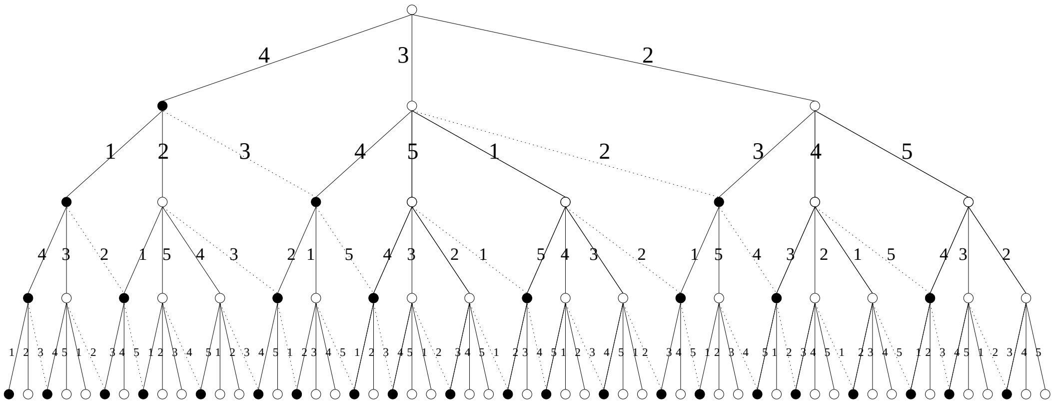

Figure 7 The Fibonacci tree with the numbering of its connections

Finally, notice that for all nodes, except those belonging to the rightmost and leftmost branches of the tree, root included, there are five edges that connect this node to others, taking into account the *parental* edge and one or two possible dotted edges, associated to missing numbers. Notice that for each node, the considered five edges are labelled from **1** up to **5** with, alternatively, clockwise and anticlockwise numberings. We shall call *connection* all the edges considered in the tree, dotted ones included.

Notice that the above numbering allows to give a different image of the tessellation, taking into account the number of each h-line that constitutes it, as shown in figure 8.

We refer the reader to [7, 8] for the formal proof of the bijection between the Fibonacci tree and the tessellation of a quarter of the hyperbolic plane.

Proving some properties of the Fibonacci tree

Let us now prove the properties we already indicated of the Fibonacci tree.

The first property is the numbering properties illustrated by figure 7 and by figure 8, where we assume all nodes numbered by the side whose reflection leads to the father of the considered node.

To that purpose, we can notice on figure 7 that the numbering of a single node is given by figure 9, below, for clockwise numbered nodes. A similar figure can be obtained for notes that are numbered anticlokwised by symmetry.

This numbering for a single node is given by the assumption we stated at the beginning of this section.

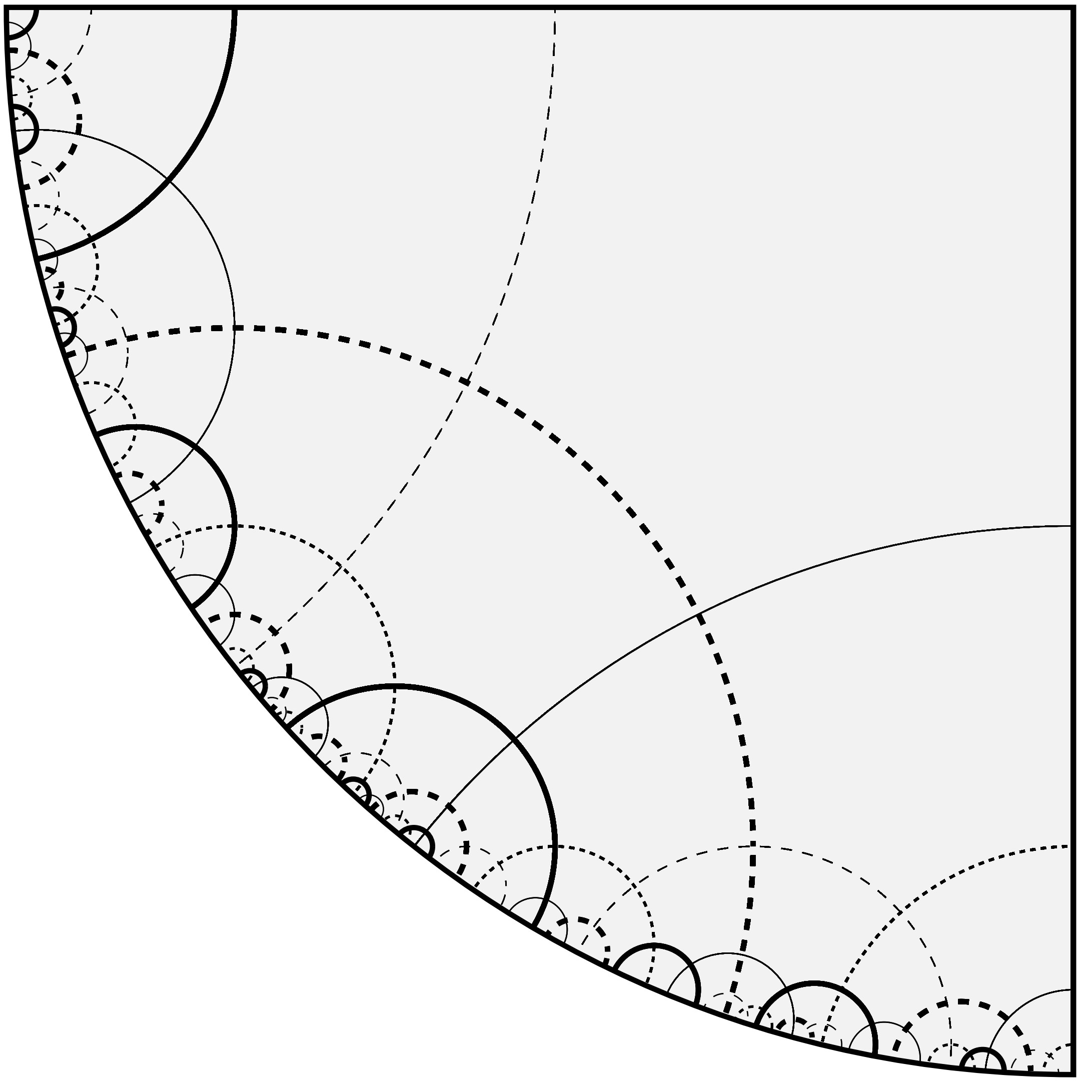

Figure 8 The tessellation in the south-western quarter
Numbering convention : solid lines : thick: 1, thin: 2; dotted lines: thick : 3, thin : 4, points : 5.

Let us now prove the property that we stated concerning the numbers on the same level.

The proof proceeds by induction on the levels of the tree. Assuming that the property is true for all nodes of the n^{th} level, we prove that it is also the case for all nodes of the $n+1^{\text{th}}$ level. To that purpose, we consider two indexes: the first one runs over the nodes of the n^{th} level and the second one runs other the sons of the node pointed at by the first index.

It is plain that the first node of a level is a 2-node and so, its first son is also a 2-node. If the father node is clockwise numbered and its own father is reached by **4**, then, according to figure 9, the considered son is reached through **1** ($\equiv$ 4+2 mod 5) and is numbered anticlokwise. The left son of that son is again a 2-node, clockwise numbered, and reached through **4** ($\equiv 1-2$ mod 5).

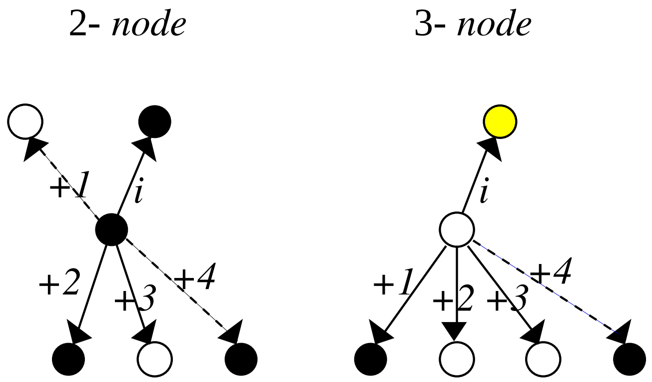

Figure 9 The numbering for clockwise numbered nodes

Assume that the first index, say **i**, points at a 2-node and that the second index, say **j**, points at the left son of this 2-node and that, according to figure 9, **j** $\equiv$ **i**+2, as far as we assume all the nodes of the n^{th} level to be clockwise numbered. Then, the numbering also matches for the right son of **i**: its number is **j+1** $\equiv$ **i+3** mod 5. Consider now the next node on the $n+1^{\text{th}}$ level. Its number should be **j+3**: when we leave a node, say **p**, a *missing* number is inserted before the number of the leftmost son of the right neighbour of **p**. So, the index of the $n+1^{\text{th}}$ level, which is presently **j+3**, points again at a 2-node, which is the leftmost son of **i-1** on the n^{th} level. But **i-1** is a 3-node and its leftmost son has number **i-1+1**. We notice that **i-1+1** $\equiv$ **(j+3)+2** mod 5, which corresponds to the insertion of the missing number. And so, the new value of the second index is indeed **j+3**. Our computation show that **j+4** matches as the second son of **i-1**. For the third son of that 3-node, we notice that **j+5** $\equiv$ **(i-1)+3** mod 5. It remains to see that the next node on the n+1$^{\text{th}}$ level has the value **j+7** $\equiv$ **j+2**, that the next node on the l^{th}evel is either a 3-node whose number is then **i-2** or a 2-node whose number is then **i-3**, since there is a missing number. The leftmost son is **i-2+1** in the first case and **i-3+2** in the second case: the same value which matches the new value of the second index, as long as **j+7** $\equiv$ **i+4**.

3. A small package for dealing with cellular automata in the hyperbolic plane

The key idea of our method is to number the nodes of the Fibonaaci tree, starting from the root, to which we assign number 1, and numbering the nodes level after level and from left to right on a same level. Then, we shall use the canonical Fibonacci of naturel numbers and with the help of some properties that we shall remind, we shall show that every information that we need about a node can be restored from its Fibonacci representation.

3.1 The Fibonacci representation of numbers

It is known that every natural number is a sum of distinct Fibonacci numbers. It is also knwon that such a representation is not unique, but it can be made unique by adding a rule, and we shall call that this new representation is the *canonical* one.

Lemma 1 – *Any positive integer is a sum of distinct members of the Fibonacci sequence.*

Proof: For the sake of completeness we recall the proof of that well-known property:

The proof is by complete induction on n, the number to be represented. And so, assume that for each $k \leq n$, k can be represented as a distinct sum of Fibonacci numbers. Consider n+1. There is a unique h such that $F_h \leq n < F_{h+1}$. As $F_{h+1} = F_h + F_{h-1}$, we obtain that $m = n - F_h$ satisfies $m < F_{h-1}$. Applying the induction hypothesis, we can represent n+1. As 1 is a Fibonacci number, our proof is complete. ■

Any distinct sum of Fibonacci with value n is called a *Fibonacci representation* of n.

Let n be a positive integer and let $n = \sum_{i=1}^{k} F_i$ be a Fibonacci representation of n. We shall represent this sum by the following word on $\{0, 1\}$: $\alpha_{k-1} \ldots \alpha_1$ where $\alpha_i = 1$ if and only if F_i occurs in the sum. By *abus de langage*, we shall also speak of that word as a *Fibonacci representation* of n.

The proof of lemma 1 provides an algorithm which allows to compute a Fibonacci representation for any positive integer. The representation yielded by that algorithm has a particluar property: among the finitely many (word) representations of the same positive integer n, there is a maximal one with respect to the lexicographic order. That representation of n is called its *canonical* Fibonacci representation. For short, we shall often say its *canonical representation*. The letters of the word are called the *digits* of the representation. We shall also fix the *positions* of the digits by numbering them from 0, which will be assigned to the rightmost digit, that is the digit corresponding to the smallest Fibonacci numbers that enters the sum constituting n.

Lemma 2 – *A representation $\alpha_k \ldots \alpha_1$ of n is canonical if and only if $\alpha_{i+1} = 1 \Rightarrow \alpha_i = 0$, for $1 < i \leq k$.*

Proof: In the proof of lemma 1 the condition $m < F_{h-1}$ entails that the condition stated in lemma 2 is statisfied. The converse is trivial: if the representation of n is canonical, the leftmost digit is 1. IF k denotes its position, this means that $F_k \leq n < F_{k+1}$. And so, by induction on k, we find again the canonical representation of n. ∎

In our sequel, we shall call *last digit* of n the digit with position 1 in its canonical representation. If $\alpha_k \ldots \alpha_1$ is the canonical representation of n, we shall say also that n is *canonically represented* by $\alpha_k \ldots \alpha_1$, and we shall denote that relation by $n \propto \alpha_k \ldots \alpha_1$. It will be also convenient to define the canonical representation in terms of the represented number and vice-versa. Accordingly, if $n \propto \alpha_k \ldots \alpha_1$, we shall write also $\overline{\alpha_k \ldots \alpha_1} = n$ and $\alpha_k \ldots \alpha_1 = \mathcal{F}(n)$.

3.2 The key lemmas

Now, we come to the lemmas which will allow us to decode the desirable information from the canonical representation of the number attaced to a node.

Definition 1 – *Let n be a positive integer. Say that n is F-odd if the last* *digit of its canonical representation is* 1 *and* F*-even if it is* 0.

Definition 2 – *Let* ν *a node of the Fibonacci tree. Say that its preferred son is the left one if nu is a* 2*-node, and the middle node if* ν *is a* 3*-node.*

Call *status* of a node ν of the Fibonacci tree the value 2 or 3 according to whether ν is a 2-node or a 3-node.

We shall give a rule which characterizes each category of nodes and we shall also prove a property about the preferred sons. We shall prove both properties together, by induction on the level of the Fibonacci tree. To that purpose, we need an auxiliary lemma about the representation of $n+1$, provided that the representation of n is given. We define the *length* of n to be the number of the digits in its canonical representation. In order to facilitate the proof of that lemma, we introduce a new operation on the canonical representations of a number.

Let $n \propto \alpha_k \ldots \alpha_1$. As a word on $\{0,1\}^*$, $\alpha_k \ldots \alpha_1$ can be split into to factors, say $\mathcal{F}(n) = w_1.w_2$. Set $m_a = \overline{w_a}$ and denote $n = m_1 * m_2$, which defines a new operation on numbers. If we impose the condition that m_1 be F-even, the concatenation of $\mathcal{F}(m_2)$ to the right of $\mathcal{F}(m_2)$ is indeed a canonical representation.

Lemma 3 – *If* $n \propto \alpha_k \ldots \alpha_1$*, then the canonical representation of* $n+1$ *is given by the following recursive rule:*

> **if** $n = m * 0$ *with* m F*-even*
>   **then** $n+1 \propto \mathcal{F}(m).1$
>   **else if** $n = m * 2$ *with* m F*-even*
>   **then** $n+1 \propto \mathcal{F}(m+1).00$
>   **else** % $n = m * 1$ *with* m F*-even*
> $n+1 \propto \mathcal{F}(m+1).0$

Proof: If n is F-even, there are two cases, depending on whether the penultimate digit is 0 or 1. In the first case, we can split n as indicated by the condition of the first **if** and the conclusion is trivial in that case. In the second case, as α_3 is necessarily 0 (it includes the case when $n = 2$, we have the splitting indicated in the condition of the second **if**. The splitting is indeed the following: $n = \sum_{i=3}^{k} \alpha_i.F_i + F_2$ and so, $n+1 = \sum_{i=3}^{k} \alpha_i.F_i + F_2 + F_1$. As $F_2+F_1 = F_3$ we have $n+1 = \sum_{i=3}^{k} \alpha_i.F_i + F_3$. If we consider now $m = \overline{\alpha_k \ldots \alpha_3}$, we see that $m{+}1 = \sum_{i=3}^{k} \alpha_i.F_{i-2} + F_1$ and so, the propagation of the carry is the same, in the case of $n{+}1$ as in the case of $m{+}1$, because of the relation $F_{i+2} = F_{i+1} + F_i$ which is invariant under any fixed shift on the indices. Accordingly, we have in that case that $n{+}1 \propto \mathcal{F}(m{+}1).00$.

Consider now the case when n is F-odd. This is the **else** branch of the second **if**. Necessarily, $\alpha_2 = 0$ in $\mathcal{F}(n)$. The same argument as in the **then** branch of the same **if** leads to the conclusion that $n{+}1 \propto \mathcal{F}(m{+}1).0.$ ■

It is now not difficult to see that from the recursive rule given in lemma 3, we can infer the following rule:

if $n \propto 1$
   then $n{+}1 \propto 10$
   else **if** $n \propto 10$
         then $n{+}1 \propto 100$
         else **if** $n \propto w10^b0(01)^a$
                  then $n{+}1 \propto w10^b010^{2a-1}$
                  else **if** $n \propto (10)^a$
                           then $n{+}1 \propto 1(00)^a$
                           else % $n \propto w10^b00(10)^a$
                                 $n{+}1 \propto w10^b01(00)^a$

Lemma 4 – *Let n be the number attached to node ν and let $n \propto \alpha_k \ldots \alpha_1\alpha_0$. Then, if m is the number attached to the preferred son of ν, then we have $m \propto \alpha_k \ldots \alpha_1\alpha_0 00$.*

Lemma 5 – *The status of a node is given by the following rule:*

if *n F-odd*
   then *status* := 3
   else **if** *$n{-}1$ F-odd*
          then *status* := 2
          else *status* := 3

Proof: We proceed by induction on the levels of the Fibonacci tree. It is plain that the property is true for the root: its canonical representation is 1, and the canonical representation of its preferred son, which is 3, is 100. The rule on the nodes is trivially true.

We know that the number of nodes on level k is F_{2k+1}. The root is F_1 and its left son is F_2. The left son of F_2 is $F_2 + F_3 = F_4$ since F_3 is the number of nodes of level 1. Assume, by induction on k, that the first node of level k is F_{2k}. As the number of nodes of level k is F_{2k+1}, the left son of F_{2k} is $F_{2k}+F_{2k+1}=F_{2k+2}$. So, we conclude that the first node of level k is F_{2k} for each k. This shows that the assertion of lemma 4 is true for the first node of level $k{+}1$ which is a 2-node. As the first node of level $k{+}1$ has 10^{2k+1} as canonical representation, the previous node, by application of lemma 3, which is a 3-node has $1(01)^{2k}$ as canonical representation and so, it is F-odd. Accordingly, the rule given in lemma 5 also applies for that node.

Assume now that the properties are true for the $j^{\rm th}$ node of level $k{+}1$, say ν_j. Consider what can be said for the $j{+}1^{\rm th}$ node, say ν_{j+1}. There are two cases: either ν_j has an F-even number or it has an F-odd one.

Consider the first case:

Let n be the number attached to ν_j. Following again the argument of lemma 3, we have two sub-cases according to whether $n=n_1*0$ or $n=n_1*2$, with n_1 an F-even number in both cases.

Consider the first sub-case:

We assume that $n=n_1*0$. Then $n{+}1\propto\mathcal{F}(n_1).1$, according to lemma 3. Let μ_j be the father of ν_j and let μ_{j+1} be the father of ν_{j+1}. If $\mu_j\neq\mu_{j+1}$, then μ_j is necessarily a 3-node and ν_j is its rightmost son. By the induction hypothesis, lemma 4 and 5 apply to μ. The preferred son of μ_j has a number whose canonical representation is of the form $w00$ and so, the number of ν_j has $w01$ as canonical representation. A contradiction with our assumption. Consequently, $\mu_j=\mu_{j+1}$ and, as a consequence, ν_{j+1} is a 3-node. As we know that its number is F-odd, lemma 5 is satisfied for ν_{j+1} in that case. Let m be the number of the preferred son of ν_j. As lemma 4 applies to ν_j, we know that $m\propto\mathcal{F}(n_1).000$. If ν_j is a 2-node, the preferred son of ν_{j+1}, which is a 3-node has number $m{+}3$, by applying definition 2. If ν_j is a 3-node and if m is the number of its preferred son, we see also that the preferred son of ν_{j+1} has number $m{+}3$. We have the following canonical representations, as n_1 is F-even:

ν_j	n	$\mathcal{F}(n_1).0$	preferred	m	$\mathcal{F}(n_1).000$
			rightmost	$m{+}1$	$\mathcal{F}(n_1).001$
ν_{j+1}	$n{+}1$	$\mathcal{F}(n_1).1$	leftmost	$m{+}2$	$\mathcal{F}(n_1).010$
			preferred	$m{+}3$	$\mathcal{F}(n_1).100$

Accordingly, the property stated by lemma 4 holds for ν_{j+1}.

Consider the second sub-case, when $n=n_1*2$:

By considering the father of ν_j, say μ_j, to which lemma 4 and 5 apply, ν_j is not a preferred son. If μ_j is a 2-node, then its preferred is its leftmost son whose canonical representation ends with 00 and so, the canonical representation of n would end with 01, a contradiction. So, μ_j is a 3-node. As ν_j is not its preferred son and as its canonical representation does not end with 01, it is a 2-node. But now, ν_{j+1} is also a son of μ_j and it is its preferred son. We have the following canonical representations:

ν_j	n	$\mathcal{F}(n_1).10$	preferred	m	$\mathcal{F}(n_1).1000$
			rightmost	$m{+}1$	$\mathcal{F}(n_1).1001$
ν_{j+1}	$n{+}1$	$\mathcal{F}(n_1{+}1).00$	leftmost	$m{+}2$	$\mathcal{F}(n_1).1010$
			preferred	$m{+}3$	$\mathcal{F}(n_1{+}1).0000$
			rightmost	$m{+}4$	$\mathcal{F}(n_1{+}1).0001$

And so, lemma 4 applies also to ν_{j+1}. From this computation we see also that lemma 5 applies to ν_{j+1} too.

Now, consider the second case, when n is F-even.

We have that $n \propto \mathcal{F}(n_1).1$, with n_1 F-even.

Considering μ_j, the father of ν_j, we conclude that ν_j is its rightmost son and, as a consequence, ν_j is a 3-node. Let m be the number of its preferred son. We know, by lemma 4, which applies to ν_j that $m \propto \mathcal{F}(n_1).100$. We also conclude that the father of ν_{j+1} is not μ_j and so, ν_{j+1} is a 2-node, as the leftmost son of a node. Consequently, its preferred son has number $m{+}2$. We have the following computation:

ν_j	n	$\mathcal{F}(n_1).1$	preferred	m	$\mathcal{F}(n_1).100$
			rightmost	$m{+}1$	$\mathcal{F}(n_1).101$
ν_{j+1}	$n{+}1$	$\mathcal{F}(n_1{+}1).0$	preferred	$m{+}2$	$\mathcal{F}(n_1{+}1).000$
			rightmost	$m{+}3$	$\mathcal{F}(n_1{+}1).001$

And so, lemma 4 applies also to ν_{j+1}. We see also that the rule given in lemma 5 applies to ν_{j+1} too.

As we examined all cases, the proof of lemmas 4 and 5 is completed. ■

Other properties can be shown about the canonical representation of the numbers attached to the nodes of the Fibonacci tree.

As an example, we give without proof the following results that the previous lemmas allow to establish.

Lemma 6 – *If ν is a 2-node and if n is its number, then $\mathcal{F}(n) = \mathcal{F}(n_1).10$, with n_1 F-even, if and only if its father is a 3-node.*

3.3 A package for the pentagrid

We provide now algorithms for obtaining the informations we need from the numbering of the nodes in the way indicated at the beginning of section 3.

All algorithms are given in a pseudo-language which allows the reader to adapt it in his/her favorite programming languages, we hope with the smallest pain. For the canonical representation of a number, for the Fibonacci sequence, or for denoting the path leading from the root to the node, we use a data structure which can be viewed as a list that can be freely scanned in both ways.

As an example, consider `canonic` which will be the name of the canonical representation used in the algorithm that we display below. It is a list of 0's and 1's. There are functions *On_Last_Of* and *On_First_Of* that allow to start the scanning of the list, either from its first element or from its last one. If we start from the first element, function *advance* gives access to the next element in the direction of the last element. If, on the contrary, we proceed from the last element, function *one step back* gives access to the next element in the direction of the first element of the list. Whatever the direction of scanning is, function *current* allows to put the currently accessed element in any variable by an affectation instruction.

For the sequence of the Fibonacci numbers, we have a file-like structure: `fibo`. It allows us to consider that we have the sequence at our disposal in that format. Function *resume* allows to go to the first Fibonacci member. Functions *advance* and *one step back* as well as *current* are also available for that structure.

We shall also use a function *Copy* when it will be needed to have a duplicate of the list in order to restore it if changes performed in a previous stage have to be cancelled.

In a first part, we give algorithms that deals with the Fibonacci tree alone. In a second part, we shall deal with the location in the whole pentagrid.

Tools for the South-Ouest Quarter

We start first with an algorithm that gives the canonical representation of a number, see below, Figure 10.

Lemma 5 gives the algorithm which is needed for obtaining the status of a node from its number. There is no need to formulate it again. We have just to notice that a function is needed that gives the last digit of the canonical representation of a number. The function consists in runing over `canonic` until its last term is met. We leave the exact formulation to the reader.

```
resume fibo
while current fibo ≤ n
loop  advance fibo  end loop
one step back fibo
n := n - current fibo
start canonic with 1
while n > 0
loop
    advance canonic
    if current ≤fibo
        then n := n - current fibo
             put 1 in canconic
        else put 0 in canonic
    one step back fibo
end loop
while present fibo
loop  advance canonic
      put 0 in canonic
end loop
```

Figure 10 An algorithm for the canonical representation of a number

Now, we formulate an algorithm for extracting the path which leads from the root to the node. The extraction is made directly from the number of the node with the help of auxiliary functions. We already know tow of them: *Status* which, from the number of the node, returns its status; and *CanRep* which returns the canonical representation of number n into the list `canonic`. We assume that we have a function *Last* which gives function: *Father*, which, from the number of a node, returns the number of its father.

We give an algorithm for *Father*, below, in Figure 11. That algorithm uses a function *Value*, which to each list $\epsilon_k \ldots \epsilon_1$ of 0's and 1's gives the number $\sum_{i=1}^{k} \epsilon_i f_i$. Below, in Figure 12, we give an algorithm that implements function *Value*.

```
canonic := CanRep(n)
On_Last_Of canonic
if  current canonic = 0
   then canonic_2 := CanRep(n+1)
        canonic := canonic_2
end if
On_Last_Of canonic
one step back canonic
one step back canonic
Father := Value(canonic)
```

Figure 11 An algorithm for computing the number of a father of a node

Indeed: if the canonical representation of n ends with 1, we have a 3-node and if μ denotes its father, in all cases, the previous node on the same level is the preferred son of μ. And so, it is enough to erase the last two digits of the canonical representation in order to obtain the canonical representation of m, the number attached to μ. The above algorithm indicates how to compute m from its canonical representation. If the canonical representation of n ends with 0, we have either the preferred son of the father, or the case when the node with number n is the 2-node of a 3-node. In the first case, we proceed exactly as we did above. In the latter case, it is enough to compute the canonical representation of n+1 which is the number of the preferred son of μ.

```
resume fibo
n_f := 0
while not Out_Of canonic
loop  if  current canonic = 1
          then n_f := n_f + current fibo
      end if
      one step back canonic
      advance fibo
end loop
Value := n_f
```

Figure 12 An algorithm for computing function *Value*

We can now obtain the algorithm for extracting the path from the root to the node if we are given the number of the node. Of course, we assume that the canonical representation of n has at least 3 digits.

```
path := ∅
canonic := CanRep(n)
On_Last_Of canonic
while n > 1
loop
    canonic₂ := Copy(canonic)
    a := current canonic
    one step back canonic
    b := current canonic
    if  a+b = 0
      then if  Status(n) = 2
                then  push 0 onto path
                else  push 1 onto path
             end if
      else  if  a =0
                then  push 0 onto path
                      m := Value(canonic₂)
                      canonic := CanRep(m+1)
                      one step back canonic
                else  if  Status(n−1) = 2
                          then  push 1 onto path
                          else  push 2 onto path
                      end if
             end if
    end if
    one step back canonic
    n := Value canonic
end loop
```

The algorithm examines the list, sarting from its last element, two digits at a time. In one step, it pushes on the top of the list which represents the path, the indication of how the path must be continued: 0 for to left, 1 for to right or to the middle, depending on the status of the node, 2 for to right in the case of a 3-node. At the end of the step, we have the number of the father of the current node and its canonical representation. Both these elements become the current values of the node number and its canonical representation. When we arrive at the root, the number is 1 and so we know that the path is completed.

Let us check the correction of the algorithm:

Indeed: when the control enter the body of the loop, $n > 1$ and `canonic` is the canonical representation of n where n is the number of the current node. As $n > 1$, there are at least two digits in the canonical representation of n. Accordingly, a and b can receive a value.

The main **if** instruction splits the examination into three cases: either $a = b = 0$, or $a = 0$ with $b = 1$, or $a = 1$ with $b = 0$, since the case $a = b = 1$ is ruled out.

If $a = b = 0$, this means that the canonical representation ends with two 0's and so, the current node is the preferred son of its father, say μ. And so, the continuation of the path starting from μ is determined by the status of the current node: if it is a 2-node, μ is also a 2-node and so the corresponding segment of the path is defined by 0; if it is a 3-node, μ is also a 3-node and the path must be continued by 1. In all cases, at the end of the body of the loop, n contains the number of μ and `canonic` contains the canonical representation of n.

If $a = 0$ while $b = 1$, we have a 2-node which is the son of a 3-node, say μ. And so the path is continued by 0. Indeed, the canonical representation of the number of a 3-node which is not a preferred son always ends with 01 because for its previous neighbour on the same level the canonical representation of the number ends with 00, whatever the status of its father μ is. Accordingly, the canonical number of μ is reached by using the canonical representation of $n{+}1$ because $n{+}1$ is the number of the preferred son of μ. The new values of n and `canonic` are, respectively, the number of μ and its canonical representation.

If $a = 1$ while $b = 0$, we have a 3-node. But his father, say μ can be a 2-node or a 3-node. In the first case, the path must be continued by 1, in the second case, it must be continued by 2. In order to determine the status of μ, the simplest way is to compute the status of $n{-}1$: it is a 2-node if and only if μ is a 2-node. In that case also we see that at the end of the body of the loop, n contains the number of μ and `canonic` contains the canonical representation of n.

We examine now the last tool we need for nodes in the South-West quarter. It is an algorithm which computes the numbers of the neighbours of the node with number n. We shall assume that the output of the algorithm is an array, say A, indexed from 0 to 5: the numbers of the neighbours are placed in $A(i)$ for i in $\{1,\ldots,5\}$. $A(0)$ contains the index of the array that contains the number of the father of the node. Index i is the number of the symmetry which transforms the pentagon attached to n into the pentagon attached to number $A(i)$.

In a first step, we assume that all the neighbouring nodes of the considered node are also in the South-West quarter.

Here is an algorithm for that purpose. It uses the previous algorithm that we shall denote as a function of the number, say *Path*.

```
list := Path(n)
On_First_Of(list)
father := 5
orientation := -1
μ := 1
while not list = ∅
loop
    edge := current list
    μ0 := μ
    if Status(n) = 3
      then case edge is
                0 : father := father+5(orientation×1)
                    μ := Preferred(μ)−1
                1 : father := father+5(orientation×2)
                    μ := Preferred(μ)
                2 : father := father+5(orientation×3)
                    μ := Preferred(μ)+1
           end case
      else case edge is
                0 : father := father+5(orientation×2)
                    μ := Preferred(μ)
                1 : father := father+5(orientation×3)
                    μ := Preferred(μ)+1
           end case
    end if
    orientation := - orientation
    advance list
end loop
```

neighbour(0) := father
neighbour(father) := μ_0
if *Status(μ_0)* = 3
  then neighbour(father$+_5$(orientation$\times$1) := *Preferred*(n)$-$1
  else neighbour(father$+_5$(orientation$\times$1) := μ_0-1
end if
neighbour(father$+_5$(orientation$\times$2) := *Preferred*(n)
neighbour(father$+_5$(orientation$\times$3) := *Preferred*(n)+1
neighbour(father$+_5$(orientation$\times$4) := *Preferred*(n)+2

where *Preferred* is a function which for each n with $n \geq 1$ computes the number of the preferred son of the node attached to n. As the computation of that function is easier than the computation of function *Father*, we leave it to the reader.

It is not difficult to show the correctness of the algorithm for all the nodes of the Fibonacci tree that have all their neighbours in the tree.

Using the path leading from the root downto the node attached to n, we compute step by step the nodes on that path as well as the index of the symmetries transforming the pentagon attached to a node into the pentagon attached to the considered son of that node. It is easy to notice that when the list is empty, $\mu = n$ and μ_0 is the number of the father of n. We have also in `father` the index of the symmetry transforming μ_0 into n. We know also the orientation of n. All these informations are enough to compute the neighbours of n. We just notice that the above algorithm uses a special addition: $+_5$. We define that operation by $a{+_5}b = k$ such that $k \in \{1, \ldots, 5\}$ and $k \equiv (a{+}b)$ mod 5, with $a, b \in \{-5, \ldots, 5\}$.

We have now to see what to do for nodes that have a neighbour out of the Fibonacci tree. It is not difficult to see that the exceptional nodes are the nodes which belong to the leftmost branch of the Fibonacci tree and the nodes which belong to the rightmost branch of that tree. The root is also exceptional as long as it belongs to both that branches and it has only three neighbours in the Fibonacci tree associated to the South-West quarter. The numbers of the exceptional nodes are the following: the Fibonacci numbers with even indices are the exceptional 2-nodes; numbers $f_{2k}-1$, with $k \geq 1$ are the exceptional 3-nodes.

Tools for the whole plane

In [8], it is indicated how to connect four Fibonacci trees in order to cover the whole plane. Taking into account the rules that we already considered for the nodes of a Fibonacci tree, we have the possibility to make the connection as smooth as possible. In order to observe the rule of the numbering which requires that for pentagons sharing an edge, the number of that edge is the same for both pentagons, we obtain the following pattern that we take from [8]:

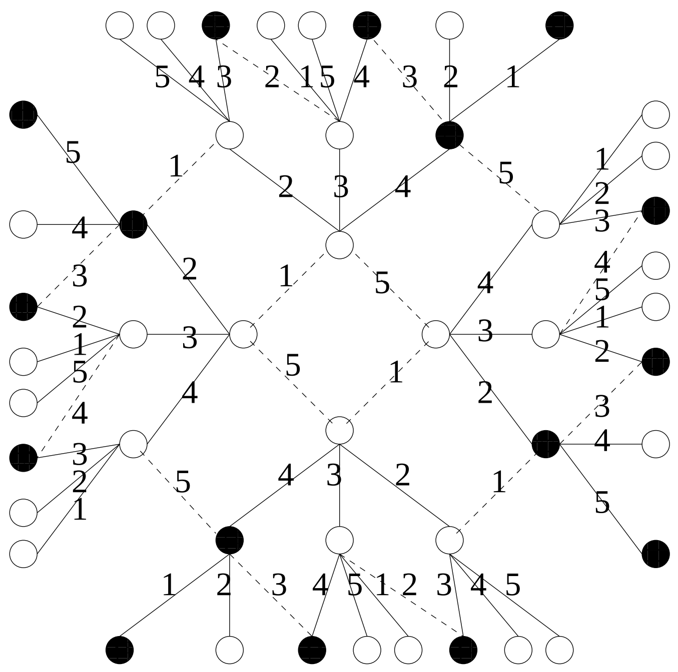

Figure 10 Connecting the four Fibonacci trees

Notice that the Fibonacci trees are obtained by composition of rotations of $\pi/2$ around the origin of the plane and in any direction. Recall that we take the center of the unit disk as the origin of the plane.

Accordingly, it is easy to show by induction on the levels that a rightmost 3-node has the leftmost 2-node of the same level of the Fibonacci tree which is obtained by an anticlockwise rotation of $\pi/2$ as its fifth neighbours. Indeed, except the root, all such nodes have their father in the same tree. Notice that our remark also settle the rule for leftmost 2-nodes on a level of a Fibonacci tree. As a consequence, we have to modify the above algorithm as follows:

During the computation of the symmetries occuring along the path, we can check whether the given node is exceptional or not. If it is exceptional, then we can compute the number of the needed node by taking the corresponding path with the same length.

Now, we have to fix a rule for the numbers of the pentagons of the whole pentagrid.

There are several ways for doing that. The first one is to split the set of positive integers according their remainder modulo 4 and to associate a positive integer to each pentagon according the following rule:

quarter	form of the numbers
South-West	$4k$
North-West	$4k+1$
North-East	$4k+2$
South-East	$4k+3$

where k is the number of the considered pentagon in its Fibonacci tree.

Another way is to use signs. Positive numbers can be given to the South quarters, negative ones to the North quarters. We can then distinguish between West and East by the ordinary parity of numbers, associating even numbers to the West quarters, odd ones to the East quarters.

Still another way is to use signs and, for instance, number i. South quarters will receive integral numbers, while North ones will receive purely imaginary ones. Mathematically, that latter solution is more elegant but, for implementation considerations, it is not simpler than the previous ones because the introduction of number i entails a special encoding.

For applications, we shall use the first encoding we considered, using remainders modulo 4.

4. Conclusion

The encoding given in this report, which is based on a simple numbering of the nodes of a Fibonacci trees, seems to us a more compact one than the representation given in [8]. Indeed, as the pentagrid impose on us the need to find the way to locate various places each one from other ones, the use of the Fibonacci tree solves the problem, as it was already noticed in [8]. The interest of the present representation is that the numbering represents an economic way to take notice of the *path* that leads from the root to a given node. Also, it provides a convenient way to represent partial configurations which can be developped from a given one by continuations according to precise rules, as for example, go along a given h-line. Such lines appear in a Fibonacci tree if we consider missing numbers: it can be noticed that the same missing number appear between the branches starting from neighbouring nodes on a same level, one branch being the rightmost branch in the sub-tree rooted on the left node, another branch being the leftmost branch in the sub-tree rooted on the right node.

As a further indication, in [8], it was indicated that the construction of the pentagrid that we recalled in this paper can be extended, almost verbatim, to regular n-gons with right angles. It can be asked whether the construction associated to the Fibonacci tree generalizes to that case. The answer is yes, and a forthcoming paper by the present author and Gencho Skordev will indicate a convenient way for performing such a program.

References

[1] Allouche J.-P., von Haeseler F., Peitgen H.-O., Skordev G., Linear cellular automata, finite automata and Pascal's triangle, Discrete Appl. Math. **66**, 1-22, (1996).

[2] C. Carathéodory. Theory of functions of a complex variable, vol.II, 177–184, Chelsea, New-York, 1954.

[3] Cattaneo G., Formenti E., Mauri G., Rule space transformations and one-dimensional cellular automata, Proceedings Second International Conference "Developments in Language Theory", J. Dassow, G. Rozenberg & A. Salomaa eds., World Scientific, Singappore, 409-419, (1995).

[4] H.W. Guggenheimer. Plane geometry and its groups. Holden Day, Inc., San Francisco, 1967.

[5] B. Iversen, Hyperbolic Geometry. Cambridge University Press, 1992.

[6] W. Magnus, Noneuclidean tessellations and their groups. Academic Press, New-York, 1974.

[7] Margenstern M., Morita K., NP problems are tractable in the space of cellular automata in the hyperbolic plane. Technical report, Publications of the I.U.T. of Metz, 38p. 1998.

[8] Margenstern M., Morita K., NP problems are tractable in the space of cellular automata in the hyperbolic plane, to appear in *Theoretical Computer Science*.

[9] Maskit. B. On Poincaré's theorem for fundamental polygons. *Advances in Math.*, **7**, 219–230, (1971).

[10] H. Meschkowski, Noneuclidean geometry, translated by A. Shenitzer. Academic Press, New-York, 1964.

[11] Poincaré H., Théorie des groupes fuchsiens. Acta Mathematica, **1**, 1–62, (1882).

[12] A. Ramsay, R.D. Richtmyer, Introduction to hyperbolic geometry. Springer-Verlag, 1995.

[13] Robinson R.M. Undecidable tiling problems in the hyperbolic plane. Inventiones Mathematicae, **44**, 259-264, (1978).

Imprimé en France Dépot légal juillet 1999 ISBN 2-9511539-4-5