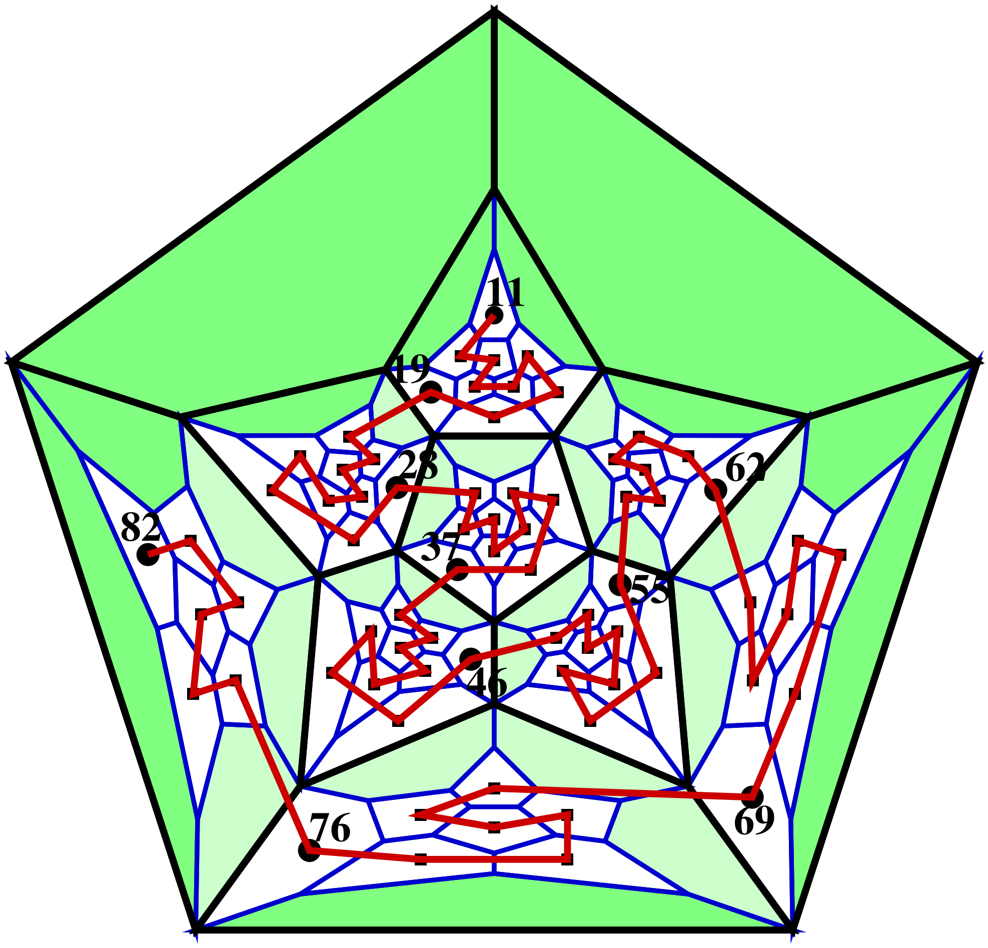


Tools for devising cellular automata in the hyperbolic $3D$ space.

Maurice Margenstern, Gencho Skordev

Publications du LITA, N° 2002-101

Service de reprographie de l'U.F.R. M.I.M., Université de Metz

Laboratoire d'Informatique Théorique et Appliquée, EA 3097 Université de Metz
Directeur de la publication : Maurice Margenstern

Tools for devising cellular automata in the hyperbolic $3D$ space

Maurice Margenstern, Gencho Skordev

Laboratoire d'Informatique
Théorique et Appliquée
EA 3097,
Université de Metz
margens@lita.univ-metz.fr

CeVis
Fachbereich Mathematik
Universität Bremen
skordev@cevis.uni-bremen.de

Abstract. The study of cellular automata (CA) on tilings of hyperbolic plane was initiated in [14]. Appropriate tools were developed which allow us to produce linear algorithms to implement cellular automata on the tiling of the hyperbolic plane with the regular rectangular pentagon.

In this paper we tackle the problem of devising similar tools in the case of the $3D$ hyperbolic space. The tools are given for the rectangular dodecahedral tiling of the $3D$ hyperbolic space. We give an algorithm which computes all needed information from the number of a cell. The algorithm is cubic in time and quadratic in space.

7, 2, 2002

1 Introduction

This paper develops tools in order to implement cellular automata (CA) on the tiling $\mathcal{T}\{5,3,4\}$ of the hyperbolic $3D$-space by regular rectangular dodecahedra. Here, we make use of the traditional *Schläfli* symbols, see [28]: 5 is the number of sides of a face, 3 the number of faces which share a common point, 4 is the number of tiles which share a common edge.

The basic ideas, namely the bijection of the tiling of a quarter of the plane with a tree and the enumeration of the nodes of that tree, are taken from [14, 15, 16, 20]. In these quoted papers the authors consider the Poincaré unit disc model of the hyperbolic plane. Here, we consider the generalisation to $3D$ of this model: the conform unit ball model. The tiling $\mathcal{T}\{5,3,4\}$ is determined by its tile $\mathcal{D}$: the regular rectangular dodecahedron. It will be enough for our purpose to restrict the $3D$ space to a spheric eigth of the space, we shall say later on an *octant*. The considred octant is delimited from the centre of the sphere by three pairwise perpendicular planes. Denote the corresponding restriction of $\mathcal{T}\{5,3,4\}$ by $\mathcal{T}$.

We construct a spanning tree T of the dual graph $\Gamma(\mathcal{T})$ of the tiling $\mathcal{T}$, *i.e.* a tree which contains all vertices of $\Gamma(\mathcal{T})$. Remind that the vertices of the last graph correspond to the tiles of $\mathcal{T}$ and that two of them are adjusted by an edge if the corresponding tiles have a common side.

The method which we use is characterised in [21]. Define a *region* of the space to be a simply connected and unbounded closed subset of $I\!H^n$, with a non empty interior. Say also that a subset X of $I\!H^n$ is a *copy* of another subset Y of $I\!H^n$ if and only if X is isometric to Y. The method goes as follows : it shows that the tiling possesses the following property, that there are finitely many regions of the space, say $\mathcal{S}_0$, $\mathcal{S}_1$, ..., $\mathcal{S}_k$, such that:

(i) each $\mathcal{S}_i$, with $i \in \{0..k\}$, can be split into a tile $\mathcal{P}$ and in finitely many simply connected unbounded closed sets where each one is a copy of some $\mathcal{S}_j$;

(ii) $I\!H^n$ can be split into finitely many copies of $\mathcal{S}_0$.

In these definitions, *splitting* means that the interiors of the considered regions are pairwise disjoint and that any point of the space belongs to at least one region. In condition (i), the isometries may be established with different $\mathcal{S}_j$'s.

Under such conditions, it is immediate to see that the splitting which we defined above, may be recursively continued. This defines a tree T: its nodes are associated with the part of $\mathcal{S}_i$ which is isometric to $\mathcal{P}$ and which is defined in the above process. There are $k{+}1$ kinds of nodes and so, the above definition gives rise to a square matrix which, for each $\mathcal{S}_i$ gives the number of copies of the set $\mathcal{S}_j$ which it contains. This induces an algebraic recurrence relation which allows us to compute the number u_n of nodes of the tree which are at the same level, see again [21].

We want to number the nodes of T in a systematic way, starting with the root, generation after generation, exactly as this was performed for planar tilings in [15] and [22, 23]. Recall that in the planar case, we also constructed a tree T_1 and we defined a numeration for the nodes of T_1. As we are in the plane, we can display T_1 by associating its nodes to the centres of the corresponding tiles. The assignment of numbers to nodes raises no problem, because in this display, a natural order is induced on the nodes by the angles between the segment which joins them to the root of T_1 and the segment which joins this root to a fixed node, for instance the leftmost son of the root. There are still many other ways to introduce such a linear object, but the one which we just mentionned is natural.

In the case of the $3D$ space, the same display of the tree T in the space, here by taking the centres of the dodecahedra, is not a planar figure and we have to project it on a plane in order to study its properties. There are a lot of possible projections and, *a priori*, none is canonical. This difficulty is well known in data compression of pictures and images, see [2] for instance.

In our first section, we shall see how a very old tool from the late 19^{th} century, *Schlegel diagrams*, comes to a valuable help. Thanks to appropriate Schlegel diagrams, we shall be able to define the numbering. Our first section will be devoted to this geometrical approach to our algorithms.

As in the planar case where the above splitting property holds, the number of nodes of T which are produced for a given generation, can be computed by a suitable recurrence relation. Together with appropriate initial conditions this relation defines a sequence of positive numbers. We expect that the greeedy representation of numbers which is associated with this sequence will help us to encode the information which we need in order to implement cellular automata in the $3D$ hyperbolic space. Recall that this is the case for the $2D$ case, see [14, 22, 23]. However, in the $3D$ hyperbolic space, this is not so easy as it is in the plane, although the computations are not that trivial neither in the plane. But in the plane, the problem is much facilitated by the fact that the language which is associated to the greedy representation is regular. In the $3D$ case, it turns out that the language which is generated by the greedy representation is not regular. It is even not context free. This will be our second section which is devoted to the proof of this result.

In our last section which is devoted to implementation, we shall present an algorithm. It works with the number of cells only. As there are several re-computations to perform in order to find the required information, the algorithm is not linear. However it is still polynomial. Indeed, it is cubic in time and quadratic in space.

2 A geometrical approach

In this section, we develop the necessary preliminaries which allow us to formulate *geometrical* algorithms in order to find the neighbours of a cell in the dodecahedral tiling, as well as to find all the cells which share a common vertex: this is a question which is raised by the re-construction of the dual graph of the dodecahedral tiling, starting from the spanning tree which we shall define in the last subsection of this section.

First, we remind the reader with elementary features of hyperbolic geometry which we shall use freely in the sequel.

Then, we introduce the main tool which will allow us *to see something* in the dodecahedral tiling. It will play a key rôle in the second section.

2.1 Hyperbolic space

We use the Poincaré unit disk model of the hyperbolic plane $I\!H^2$, [3, 24].

Remind that in this model, the straight lines, also called h-lines, are the intersections with the disk of the circles which are orthogonal to the unit circle ,the boundary of the unit disk. In particular all diameters of the disk are h-lines. The angles between the h-lines are the euclidean angles between the circles which support them. The points of the unit circle are called *points at infinity* for $I\!H^2$.

Let k be a circle orthogonal to the unit circle. The inversion with respect to k which is restricted to $I\!H^2$ is a hyperbolic rigid motion: the reflection of $I\!H^2$ in the h-line being supported by k. When k is a diameter, we consider the usual reflection in k which is a limit case of inversions with respect to a sequence of circles orthogonal to the unit circle which would converge to k. All hyperbolic motions are products of finitely many reflections.

The model can be extended to three dimensions, more generaly to any n dimensions, by considering the open unit ball, respectively hyperball. Denote this space by $I\!H^3$, see [1]. Planes are now the intersection in $I\!H^3$ of spheres which are orthogonal to the unit sphere, the border of $I\!H^3$. Notice that diametral planes are a particular case. Notice also that, by our definition, points of the unit sphere do not belong to $I\!H^3$. As in the case of $I\!H^2$ with the unit circle, we call the points of the unit sphere *points at infinity* for $I\!H^3$. Lines are intersections of secant planes. As the considered spheres are orthogonal to the unit sphere, the plane which contains their intersection is diametral.

This is easily proved by considering the images of a point of the intersection of these planes under the reflection in each plane: the images also belong to the intersection, which means that the euclidean line which passes through them also passes through the centre of the unit ball.

Now, notice that a diametral plane provides us with a copy of $I\!H^2$. And so, the considered line is also a line in this copy of $I\!H^2$. Consequently, a line has two points at infinity and, conversely, two points at infinity define a line which is determined by the diametral plane which contains these points.

A lot of properties of the planar case can be transported into the $3D$ hyperbolic space. As an example, non-secant planes have a single common perpendicular. However, there are specific properties of $I\!H^3$ in this regard. In particular, whereas there are infinitely many tilings which are based on regular polygons in $I\!H^2$, there are only finitely many ones in $I\!H^3$ which are based on a regular polyhedron, namely: $\mathcal{T}\{5,3,4\}$, $\mathcal{T}\{5,3,5\}$, $\mathcal{T}\{4,3,5\}$ and $\mathcal{T}\{3,5,3\}$. Among them, a single one is rectangular, namely $\mathcal{T}\{5,3,4\}$, see for instance [28, 5, 30]. Its faces are rectangular regular pentagons and the dihedral angle between faces is the right angle. There is however another dodecahedral tiling of $I\!H^3$, with other angles, $\mathcal{T}\{5,3,5\}$, see also [5]. The reader may find other information and references in [28, 6]. There is also an internet site to be visited: *http//:*.

We choose the rectangular dodecahedral tiling which, in some sense, generalises the pentagrid of $I\!H^2$, see [**?**], to $I\!H^3$.

The first idea is to apply to $\mathcal{T}$ the methods which were applied to the pentagrid in [**?**] and which we recalled in the introdution. We shall see that it is possible, but the results are surprisingly significantly different: three dimensions cannot be reduced to two ones, also in the hyperbolic world.

2.2 Schlegel diagrams

Let us now introduce a foundamental visualization tool of the dodecahedron: *Schlegel diagrams.*

The Schlegel diagram of a figure F in a $3D$ space, either euclidean, elliptic or hyperbolic, is the central projection of F on a plane, starting from a particularly chosen point of the space. For more details on these diagrams, see for instance, [28, **?**, 10]. Such a representaton takes place in the euclidean plane but, as already indicated, it can be used for the study of $3D$ spaces, whatever the geometry is.

Recall that in $I\!H^3$, there are no similitude, as this is also the case in $I\!H^2$. Consequently, all rectangular dodecahedra of $I\!H^3$ are pairwise isometric. And so, in some sense, there is a single dodecahedron from the point of view of the Schlegel diagram. We shall project dodecahedra on their *bottom* plane, starting from a point which is near the opposite face, out of the convex hull of the dodecahedron.

We shall denote by $\mathcal{D}$ the rectangular dodecahedron with one vertex at O, the centre of the unit ball. The three faces which share O are supported by planes which are pairwise perpendicular. We shall decide that one of these planes supports the *bottom* face of $\mathcal{D}$. The two other planes will be called the *walls* of the dodecahedron $\mathcal{D}$.

In Figure 1, we have a picture of Schlegel diagram of $\mathcal{D}$ as well as of any dodecahedron.

In the figure, we numbered the faces from 1 until 11. There are indeed twelve faces which are represented. The pentagon which encloses all the numbered faces is also a face. Let us see it as the bottom face of $\mathcal{D}$ and let us number it by 0. We also shadowed the walls for the following reason. As indicated in the introduction, we restrict our attention on the part of $\mathcal{T}\{5,3,4\}$ which lies in the octant $\mathcal{O}$ which is delimited by the three diametral planes, pairwise perpendicular, which we selected for the bottom and the walls of $\mathcal{D}$. The shadowed faces represent faces which are supported by the walls of $\mathcal{D}$. Considering the sides of the bottom pentagon, we shall call *free*, the sides of this pentagon which do not belong to a shadowed face. We shall also say that the sides of this pentagon are the *outer* sides of $\mathcal{D}$.

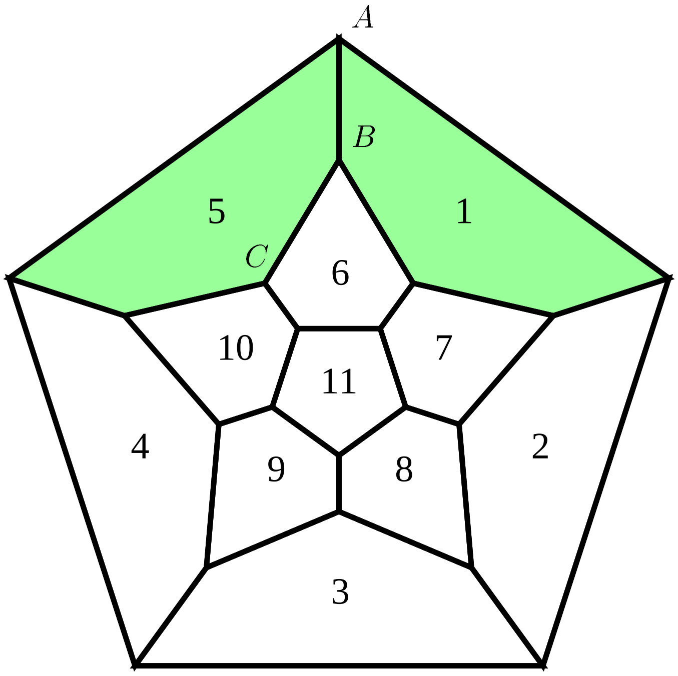


Figure 1. A Schlegel diagram of $\mathcal{D}$.

We shall extend to $\mathcal{O}$ the notion of *bottom* and *walls* which we introduced for $\mathcal{D}$.

2.3 Splitting the octant

Our first goal will be to split the octant by applying the method which is given in [21] and which we recalled in the introduction. Notice that here, the regions will be intersections of finitely many half-spaces: call them *simple sets*. Notice that the octant $\mathcal{O}$, as any octant, is a particular case of a simple set: it is defined by its walls and its bottom.

We have here to notice several points.

First of all, we interpret the diagram of Figure 1 both as a representation of the dodecahedron $\mathcal{D}$ and as a representation of the octant $\mathcal{O}$ which is defined by the bottom and the walls of $\mathcal{D}$. Accordingly, the diagram represents also an *infinite* object.

We shall say that $\mathcal{D}$ is the *leading* dodecahedron of the octant $\mathcal{O}$. In this regard, walls must be seen as the restriction to $\mathcal{O}$ of the whole planes which support the corresponding faces of $\mathcal{D}$. In view of the splitting process which we shall define, walls must also be seen as *obstacles*: this means that it is forbidden to reflect $\mathcal{D}$ in a face which is supported by a wall. It is also forbidden to reflect $\mathcal{D}$ in the bottom. Indeed, reflection of $\mathcal{D}$ in the bottom or in a wall would yield a dodecahedron which is not included in $\mathcal{O}$ and hence, which does not belong to $\mathcal{T}$. By contrast, the other faces of $\mathcal{D}$, which are in white colour in Figure 1, will be called *white faces*. It is always possible to reflect $\mathcal{D}$ in a white face. In the sequel, the name or the number which is given to a face will also be used to denote the plane which supports this face. Walls and bottoms will also create *shadowed* faces. The shadowing process, which we shall soon explain, will play a key rôle in our sequel.

Let us say that $\mathcal{D}$ is the root of the tree whose construction we perform at the same time as the splitting. Say that $\mathcal{D}$ belongs to the 0^{th} generation.

The splitting is a recursive process which is performed *generation* after *generation*.

The first generation of the splitting is obtained as follow: we want to split the closure of the complement of $\mathcal{D}$ in $\mathcal{O}$ into simple sets, according to what is defined in our introduction. We shall try to extract as many octants as possible.

In the complement of $\mathcal{D}$, we can define an octant $\mathcal{O}'$ by taking a face as the bottom of $\mathcal{O}'$ and by taking two adjacent faces with a common point with the bottom as walls for $\mathcal{O}'$. As an example, this is immediately the case of the face 6 on $\mathcal{D}$. Now, notice that planes 5, 1 and 6 define a new octant, $\mathcal{O}_6$. All octants are isometric and it is easy to define an isometry which transforms $\mathcal{O}$ into $\mathcal{O}_6$. It is the shift along the line AB which transforms A into B. Denote by $\mathcal{D}_6$ the image of $\mathcal{D}$ by reflection in the face 6 which is white. We notice that $\mathcal{D}_6$ is in the same relation with $\mathcal{O}_6$ as $\mathcal{D}$ is with $\mathcal{O}$. For this reason, we say that $\mathcal{D}_6$ is the leading dodecahedron of $\mathcal{O}_6$. As planes 5 and 1 are walls for $\mathcal{O}_6$, we shall also denote $\mathcal{O}_6$ by (6; 5,1), where we first indicate the bottom and then the walls. This notation will be used for all the octants which we shall define and it will easily be extended to any simple set which we shall define a bit later. As an example $\mathcal{O}$, is denoted by (0; 5,1).

Now, the plane 6, which is the bottom of $\mathcal{O}_6$, allows to delimitate a new octant with the plane 5 as its walls and with the face 10 as its bottom: (10; 5,6). Its leading dodecahedron is the reflection of $\mathcal{D}$ in the face 10, the dodecahedron $\mathcal{D}_{10}$. It is also obtained from $\mathcal{D}$ by the product of two shifts: first from A to B along AB, then from B to C along BC.

Of course, if we reflect $\mathcal{D}_{10}$ in the face 10, we obtain again $\mathcal{D}$. So, the interdiction which is associated to the bottom is natural, as it is for walls. Why is it also natural to forbid reflections of $\mathcal{D}_{10}$ in its walls?

In order to see that, take a cut of the dodecahedron in the plane which bissects the common side of faces 6 and 10. Figure 2, below illustrates what happens.

Figure 2 shows that the reflection of $\mathcal{D}_6$ in the support of the face 10 is also the reflection of $\mathcal{D}_{10}$ in the support of the face 6. And so, the shadowing of faces which we shall soon state is a rule to fix a choice between the two possible reflections: one is forbidden, the other must be performed. In particular, this means that the continuation of the face in which the reflection is permitted becomes a wall for the other face.

The process of shadowing is performed face after face, each time when a new dodecahedron is produced by reflection in the considered face. In our representation, the completion of the reflection is indicated by developing the Schlegel diagram of the resulting new dodecahedron. This diagram is inscribed in the face in which the reflection is performed and which appears to be the bottom of the new dodecahedron. The faces of the diagram belong to the next generation in comparison with the generation of the bottom of this new dodecahedron.

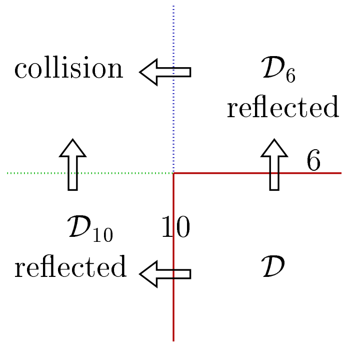


Figure 2. Why the shadow principle.

And so, we obtain the first generation of dodecahedra by taking the successive reflections of $\mathcal{D}$ in its *white* faces.

We already noticed that the first dodecahedron which is produced is $\mathcal{D}_6$, see the upper left hand picture of Figure 3. We also noticed that planes 5 and 6 are walls for the face 10 which is a bottom. And so, (10; 5,6) is our next octant, see the upper right hand picture of Figure 3. The pictures of Figure 3 illustrate how the process goes on. Now, planes 6 and 10 are walls for the face 11, which yields the octant (11; 6,10). But now, planes 10 and 11 are walls for the face 9, which gives us the octant (9; 10,11). Similarly, planes 9 and 11 are walls for the face 8 and so, we get the octant (8; 9,11).

But now, the face 7 is surrounded by four walls: planes 1, 6, 11 and 8. This is a new kind of simple sets: the *corridors*. We denote them by $(a; b, c, d)$, where a is the bottom of the dodecahedron, and b, c, and d are planes which are orthogonal to a and which pass through sides of the pentagon which is associated to a. The dodecahedron D which belongs to the intersection of the half-spaces being defined by a, b, c and d and which contain D, is called the *leading dodecahdron* of the corridor. At the present moment, we got the corridor (7; 1,6,11,8).

Next, we reach the outer faces of $\mathcal{D}$: as the plane 7 is a wall for the face 2, we get that (2; 8,7,1,0) is a corridor, where 0 is the bottom of $\mathcal{D}$, which is a wall for the plane 2. Now, the plane 2 is a wall for the face 3 and so we obtain that (3; 9,8,2,0) is also a corridor. This time, the plane 3 is a wall for the face 4. It is easy to see that all sides of this face belong to a wall for the face. And so, we get a new kind of simple set which we call a *tunnel*: (4; 5,10,9,3,0), using the natural extension of the notation which we used up to now. We shall say also that the dodecahedron $\mathcal{D}_4$ is the *leading dodecahedron* of the tunnel.

At this point, just notice that our construction induces a *natural* order on the faces, provided that we decide to chose the face 6 as a sarting point and then the face 10 as the second step. The order is:

$$6,\ 10,\ 11,\ 9,\ 8,\ 7,\ 2,\ 3,\ 4$$

and we call it the *direct* order.

The choice of face 6 is somehow natural, by symmetry reasons. On another hand, the choice between 10 and 7 as a second octant is arbitrary. If we go on with 7, we obtain another order:

$$6,\ 7,\ 11,\ 8,\ 9,\ 10,\ 4,\ 3,\ 2$$

which we call the *reverse* order.

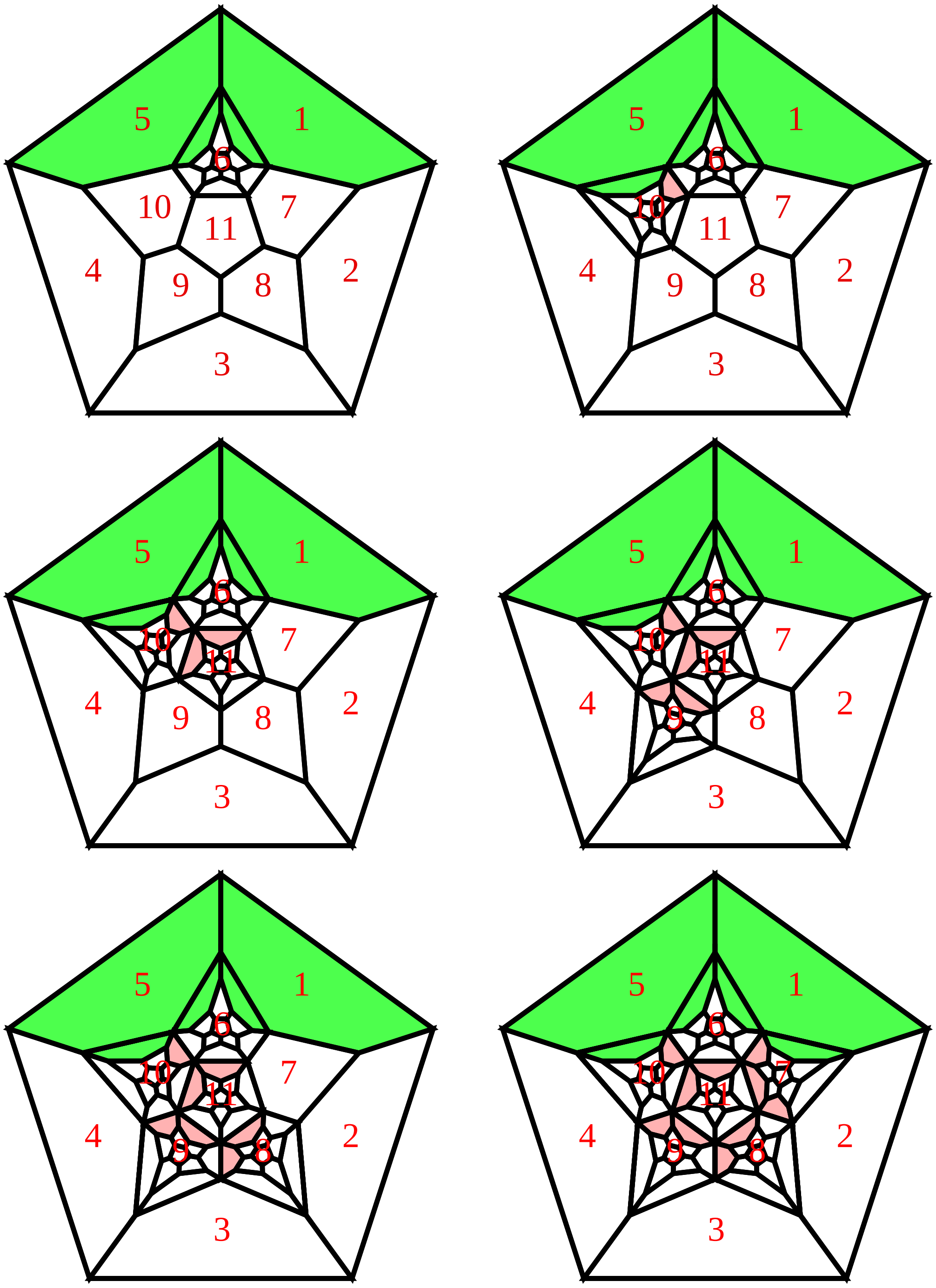
We shall go back a bit later on the choice of the face 6 as a starting point in a splitting process.

It is now easier to explain the continuation of the process.

Recall that in a Schegel diagram, there are two rings of five pentagons. Call the first one the *outer* ring and, in our numbering, it consists of the faces 1, 2, 3, 4 and 5. Call the other ring the *inner* ring, and it consists of the faces 6, 7, 8, 9 and 10. Notice that the sides which circumscribe the outer ring define the bottom face. Similarly, the sides which are inscribed in the inner ring define the face which is opposite to the bottom: the top of the dodecahedron. The numbering which we just used will be called later the *local* numbering or *local coordinates* of the faces of the considered dodecahedron. Later we shall introduce a *global numbering* and we shall ten indicate the correspondance with the local one.

As a first remark, notice that a possible wall or bottom is generated by a white face of a Schlegel diagram which is in the outer ring of faces of the dodecahedron. The outer sides can be divided into *free* sides and *barred* ones: the former are sides of a white face, while the latter belong to a wall.

As a second remark, notice that we can introduce the notion of *generation* for the dodecahedra of $\mathcal{T}$. By definition, $\mathcal{D}$ is the single dodecahdron of the 0^{th} generation. Once the n^{th} generation is completed, the dodecahedra of the $n{+}1^{\text{th}}$ generation are obtained by reflecting each dodecahedron of the n^{th} generation in all its white faces. The shadowing process allows us to introduce an order on the dodecahedra of the same generation. We shall use this order to enumerate the dodecahedra.


5
1
6
10
7
11
4
2
9
8
3

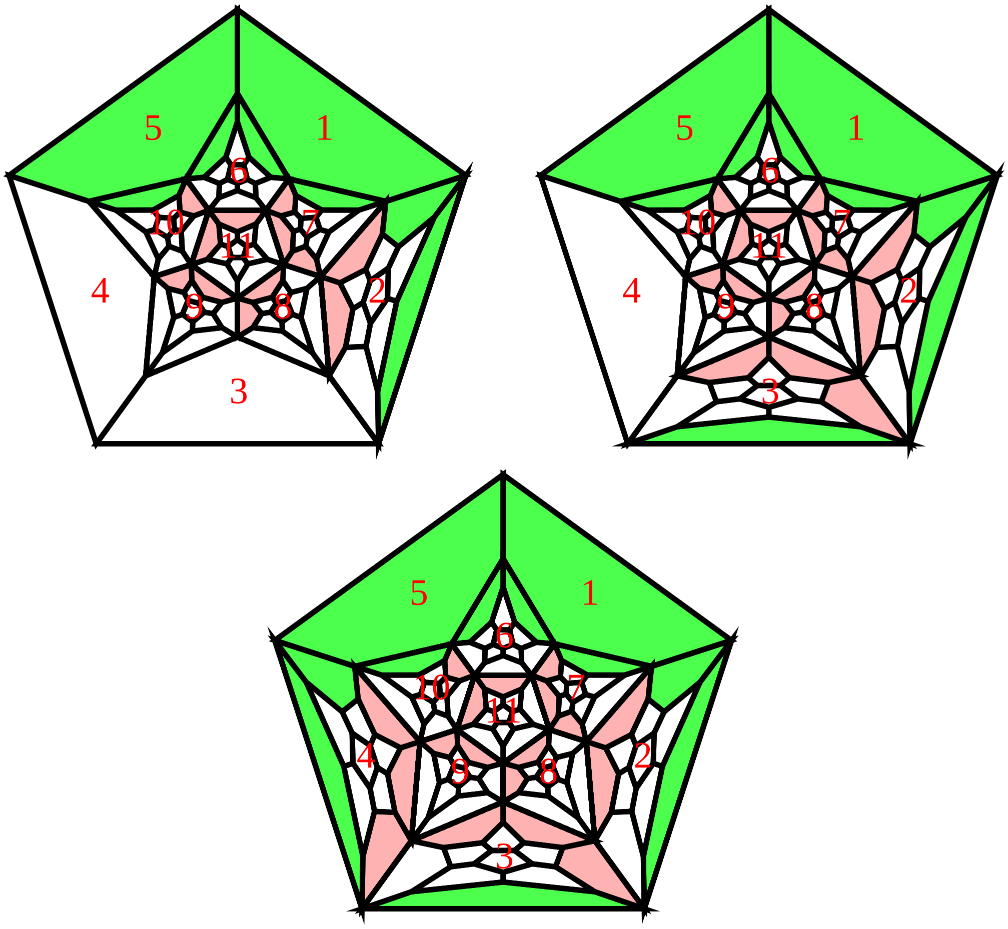


Figure 3, the shadowing process

Recall that we denote by $\mathcal{D}_\nu$ the dodecahedron with ν as its number. We shall say that the white faces of a dodecahedron of the n^{th} generation belong to the faces of the n^{th} generation.

Consider a face F of the n^{th} generation and develop the Schlegel diagram of the dodecahedron Δ_F which is obtained by the reflection in F of the dodecahedron which supports F. The white faces which are defined in this Schlegel diagram after the shadowing process do belong to the $n{+}1^{\text{th}}$ generation of faces. Assume that F is the last face producing a dodecahedron at the stage of the shadowing process under consideration. Let G be the next face to be processed. The faces F and G have a common side σ. Now, F

is the bottom of Δ_F and the plane of F is a wall for G. Consider now Δ_G which is obtained by the reflection in G and develop its Schlegel diagram which is inscribed in G. The outer face of Δ_G which has σ as a side must be shadowed: apply the argument which corresponds to Figure 2 to F and G.

It is convenient now to remember that we had two reasons to bar the reflection of a face. It is worth to distinguish between them as we shall see a bit later, when it will be the case to determine the neighbours of a cell. Accordingly, we shall represent by a dark shadowing the faces which are supported by a wall or by the bottom of $\mathcal{O}$, and we shall call them *dark faces*. The other faces in which the reflection is barred, are called *grey* faces. Accordingly, a shadowed face is either a dark one or a grey one.

Presently, we come to another point, which is a consequence of the shadowing process.

Consider again two faces F and G of the n^{th} generation which have a common side σ, and assume that F is processed before G. This means that just before the time when G is processed, the Schlegel diagram which is based on F has been drawn. It is a situation which is similar to what is depicted by one of the pictures of Figure 3. Let H be the sub-face of F which is adjacent to σ. When the Schelgel diagram which is inscribed in G is developped, one of the sub-faces of G is adjacent to σ: call it K. As F is processed before G, H is white and K is grey. According to our previous convention, K cannot be dark. Notice that H is a face of the $n{+}1^{\text{th}}$ generation. Consider now the time when H is processed. The Schlegel diagram which is then developped produces faces of the $n{+}2^{\text{th}}$ generation. Let H_1 the sub-face of H which is adjacent to σ. The argument of Figure 2 tells us that H_1 cannot be white: replace for instance 6 by F and 10 by G. On this picture, we see that H continues G: H is in the same plane as G. The dodecahedron which is produced by the reflection of Δ_F in H shares a face with Δ_G: it is namely H_1. This means that H_1 and K represent the same face. As K is already indicated and marked as a wall, we decide to colour H_1 in *black* and we say that H_1 is a *black face*.

This process of *identification* of faces goes further. It is illustrated by Figures 4 and 5.

In these figures, F and G are assumed to belong to the same generation and they share a common side σ. The sub-faces of consecutive Schlegel diagrams are numered 0,2,4, ... inside F and 1,3,5, ... inside G. In Figure 4, 0 is the face which previously we called H, while 1 corresponds to what was called K. The sides which are opposite to σ in 1 are shared by sub-faces of G which belong to the inner ring of its Schlegel diagram. Now, the sub-faces of a Schlegel diagram which belong to its inner ring as well as the top of the corresponding dodecahedron are always white faces. Accordingly, the sub-face 3 is a sub-face of such a face L and it belongs to the outer ring of the Schelgel diagram of L. Similarly, if the Schlegel diagram of 0 is developped, let 2 be its outer face which has σ as a side. We know that 2 is a black face. But the adjacent inner faces of 0 are white. The cut along AB which is performed on Figure 4 and which is represented in Figure 5 allows us to see that 3 is a continuation of 1 in the plane of 1. Call M the inner face of 0 which is adjacent to 2 and which is cut by AB. The face 4 is the outer face of the Schlegel diagram of M which is adjacent to 2. It is not difficult to see that the faces 4 and 3 are identical. By induction on the number of generations, it is easy to assume that the shadowing process possesses the following property: if a face F is processed before a face G of the same generation, the white sub-faces of F are all processed before the white sub-faces of G. Accordingly, as L is processed before M and as 3 must be grey, 4 must be black.

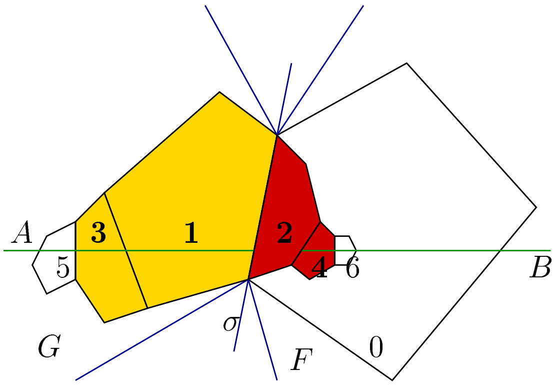


Figure 4. The mirrorship between grey/black faces.

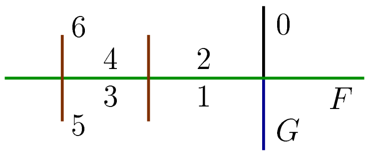


Figure 5. The cut along AB.

The Figures 4 and 5 represent a "half"-step further in the shadowing process: The face 5 is an inner face of L and the face 6 is an inner face of M. Both are white faces and, as it is explained in Figure 5, if we develop the Schlegel diagram which are respectively inscribed in faces 5 and 6, the corresponding dodecahedra share a face. Consequently, they are neighbours.

As the shadowing process is an infinite process, this situation can be generalised and we shall do it later, in the sub-section which is devoted to finding the neighbours of a cell.

For the present moment we can state:

Rules of adjacency:

Let F be a white face of the n^{th} generation in which the reflection produces a new dodecahedron $\mathcal{E}$ of the $n{+}1^{\text{th}}$ generation which we identify with its Schlegel diagram which is inscribed in the face F with i the same outer sides. Then:

- *the outer sides of $\mathcal{E}$ are the sides of F;*
- *let s be an outer side of $\mathcal{E}$; it is shared by an inner face I of $\mathcal{E}$ and a face K which is distinct from F but which is of the same generation as F;*

  then:
  - *if K is white, then I must be grey;*
  - *if K is dark or black, I must take the same colour;*
  - *if K is grey, let G be the face of the $n{-}1^{\text{th}}$ generation which contains F; then I is grey if K is in G, otherwise it is black.*

We have now to complete the description of the splitting of $\mathcal{O}$.

Our first step has led to the splitting of $\mathcal{O}$ into nine simple sets of three kinds. From now on, we identify the simple set under consideration with its leading dodecahedron and the leading dodecahedron with its bottom. In such a way, as octants are of the form $(a;\ b,c)$, we shall call them *faces of type* 3, because three of their outer sides are free. For the same reason, corridors, which have the form $(a;\ b,c,d,e)$, are called *faces of type* 1. Similarly, tunnels, which have the form $(a;\ b,c,d,e,f)$, are called *faces of type* 0.

We already indicated that the shadowing process is infinite and we have seen the principles which allow us to ground the adjacency rules which we applied in the shadowing of an octant.

As we have already seen for the second generation, the splitting of an octant leads to new octants and also to new simple sets. We found two new kinds of them in splitting an octant: the corridors and the tunnels. The splitting of a corridor may be done without raising new kinds of simple sets. This is not the case for tunnels. Their splitting necessarily introduces a newcomer: the *half-octant*, which is defined by the occurrence of two consecutive free sides on the outer pentagon which delimits the Schlegel diagram of the considered region. As it is indicated on Figure 6, where the four possible cases are represented, splitting the half-octant does not raise new kinds of simple sets. In our notations, a half-octant is denoted by $(a;\, b,c,d)$, where a is its bottom and b, c and d are its walls.

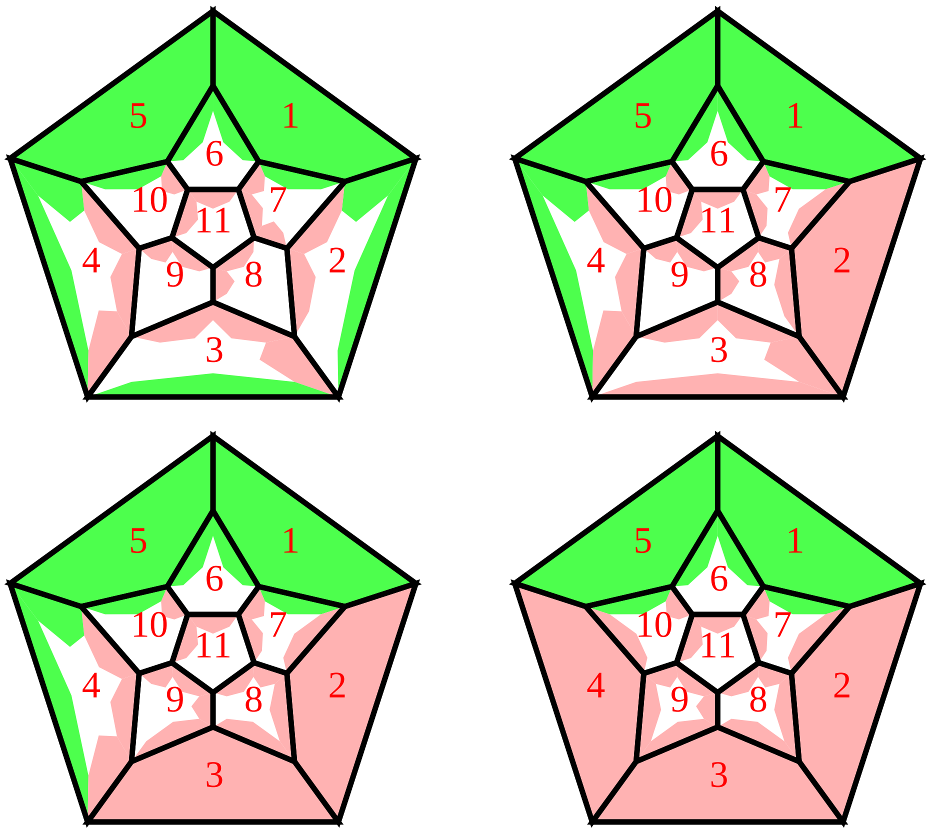


Figure 6. Splitting regions:
above, to left: the octant; to right: the half-octant;
below, to left: the corridor; to right: the tunnel.

Now, with these indications, we may apply the splitting, recursively, up to any generation, making use of the information which is provided by Figure 6 and by applying the adjacency rules.

It is not difficult to show that, as the rank of the generation which we consider is higher, the euclidean diameter of the considered regions decreases. By repeating the argument which was given in [14], it can be shown that the decreasing is exponential and so, the same argument as in that paper shows that the splitting is in a one-to-one correspondance with the tiling $\mathcal{T}$.

2.4 The spanning tree of the dual graph of $\mathcal{T}$

We associate a node with the leading dodecahedron of each region being generated in the splitting. The process starts from the root, which is associated to the leading dodecahedron of $\mathcal{O}$: the dodecahedron $\mathcal{D}$.

We already identified simple sets and leading dodecahedra by the bottom face of the dodecahedron. We shall also identify the faces, in this sense, by the node of the tree in which they are in correspondance by the splitting process. We shall use indifferently the words *face* and *node*. We shall also use the word *cell* to denote the same objects. Accordingly we say also that the node which corresponds to a region whose bottom is a face of type i is a node of type **i**. We shall also say i-node for short.

The sub-figures of Figure 6 can be translated into rules for the construction of a tree. The rules can be expressed in the form of rewriting rules, which indicate the types of the nodes which are generated:

$$\begin{array}{l} \mathbf{3} \longrightarrow \mathbf{3},\, \mathbf{3},\, \mathbf{3},\, \mathbf{3},\, \mathbf{3},\, \mathbf{1},\, \mathbf{1},\, \mathbf{1},\, \mathbf{0} \\ \mathbf{2} \longrightarrow \mathbf{3},\, \mathbf{3},\, \mathbf{3},\, \mathbf{3},\, \mathbf{1},\, \mathbf{1},\, \mathbf{1},\, \mathbf{0} \\ \mathbf{1} \longrightarrow \mathbf{3},\, \mathbf{3},\, \mathbf{3},\, \mathbf{1},\, \mathbf{1},\, \mathbf{1},\, \mathbf{0} \\ \mathbf{0} \longrightarrow \mathbf{3},\, \mathbf{3},\, \mathbf{2},\, \mathbf{1},\, \mathbf{1},\, \mathbf{0} \end{array} \qquad (R)$$

These rules give the information being contained in Figure 6 in a synthetic way but partially. Clearly, these rules allow us to define a tree by induction. The root is a node of type **3** and later, we generate the next level by applying the appropriate rule to each node of the current level. A level in the tree corresponds exactly to a generation of the splitting. Let us denote T the tree which is obtained by this process.

It is not difficult to see that we have the following result:

Theorem 1 – *The nodes of the tree T are in a one-to-one correspondance with the tiles of $\mathcal{T}$, the root being associated to $\mathcal{D}$, i-nodes to faces of type i,* *with $i \in \{3,2,1,0\}$, following the rules (R) which define T. Also, T is the spanning tree of the dual graph of $\mathcal{T}$.*

Proof: T is the spanning tree of the graph because it contains all the vertices of the graph if we identify the nodes of the tree with the vertices of the dual graph $\blacksquare$.

The rules (R) define T as a *planar* tree and they display the sons of a node by starting with the highest type and then turning to the next one, in decreasing, when we have finished with the last son of the currently highest type.

However, as we shall see in the third section, this representation is not enough to provide feasible algorithms. Figure 6 contains more information than the one given by the rules (R).

Accordingly, we go on to study the geometrical aspect of our problem. We shall deal successively with two of them: locating the neighbours of a cell and finding how to reconstruct the dual graph of $\mathcal{T}$ from its spanning tree T.

2.4.1 The neighbours of a cell

The problem of finding the neighbours of a cell in $\mathcal{T}$ is not at all immediate. The way in which we use Schlegel diagrams provides us with a *euclidean* and *planar* representation of $\mathcal{T}$. Also, as we already noticed it, these diagrams represent an infinite space within a finite volume. For these reasons, we introduced important distortions in the figures and so, our Schlegel diagrams need to be interpreted. The shadowed faces which we introduced by the rules of adjacency in section 2.3 are responsible of these distortions. In particular, black faces correspond to a partial *destruction* of parts of the planar figure: the black faces have automatically grey counterparts in the same Schlegel diagram. In order to get a bijection, black faces should be seen as destroyed. Accordingly, this prevents us from a direct look of the geometrical relations inside the tiling. However, with not too much efforts, it will be possible to define the neighbours of a given cell.

First, Figure 7, below, illustrates a representation of the second generation of the tiles of $\mathcal{T}$. On the figure, for each white face of the first generation, we developped the Schlegel diagram of the dodecahedron which is obtain from $\mathcal{D}$ by reflection in the face. We followed the convention which was introduced in section 2.3: developping a Schlegel diagram within a face means that we performed the reflection in the correpsonding face. In particular, the face is now *hidden*. This is another aspect of our diagrams: indeed, if we consider that a generation is produced in one top of a fixed clock, we need an infinite time to complete the diagram. Accordingly, we have only *partial* diagrams: up to a given number of generations. In this process, we can see the tiling as an oignon with an infinite number of skins, each one corresponding to a generation, which are put one after another, revealing new details but also hiding previous ones.

Again, we identify the tiles of the tiling with the cells of a cellular automaton which is constructed on the tiling and with the nodes of the tree which is in bijection with the tiling. And so, we shall indifferently say node n, cell n, tile n and face n for any of these objects to which the number n is attached.

From what we already know, the sons of a node n represent a part of the neighbours of that node. Clearly, the father of a node is also a neighbour. Now, a cell has twelve neighbours. And so each shadowed face F also gives access to a neighbour: it is the cell which is 'behind' F, but it cannot be represented at this place because, due to the shadowing, the reflection in F is namely forbidden. And so, the problem consists in finding where is the diagram of the dodecahedron which also occupies F. In short, as we shall see, the neighbour in F in the diagram is the *basic* cause of the shadowing of F.

In order to clarify that point, we have to examine all the different cases.

Presently, we generalise the situation which is depicted by the Figures 4 and 5 thanks to the notion of a *face line*:

First, we define a function on the set of faces, *gen* which, to a given face F, returns the number of its generation. By definition, if F is a face of $\mathcal{D}$, $gen(F) = 1$.

Consider a sequence $F_1, \ldots, F_k, F_{k+1}$ of faces such that:

- (i) $gen(F_{i+1}) = gen(F_i) + 1$ for $1 \leq i < k$;
- (ii) $gen(F_{k+1}) = gen(F_k)$;
- (iii) for $1 \leq i \leq k$, there is a face G_i with $gen(F_i) = gen(G_i)+1$ such that F_i is an outer face of G_i; moreover, F_i is a grey face and F_{i+1} is not contained in F_i;
- (iv) F_i and F_{i+1} are adjacent for $1 \leq i \leq k$;
- (v) F_{k+1} is white.

We say that $F_1, \ldots, F_k, F_{k+1}$ is a *grey stack*, that F_{k+1} is the *top* of the stack, that k is its *length* and that F_1 is its bottom.

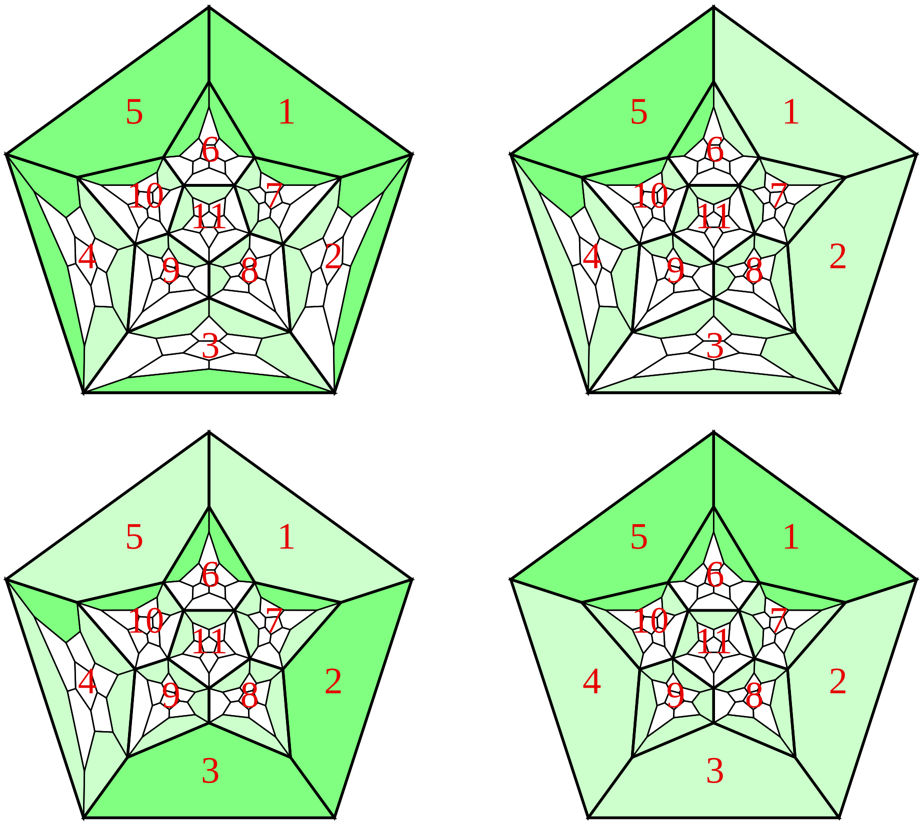


Figure 7. The second generation for each type of face.

The position of F_{i+1} with respect to F_i can be defined by an integer in $\{0,1,2,3\}$ in the following way: these numbers are attached to the sides of F_i which are not shared by F_{i+1} and they number these sides anticlockwise. Accodingly, we associate a number ϵ_{i+1} to F_{i+1} which is called the *index* of F_{i+1} with respect to F_i: it is the number of the side which is common to F_i and F_{i+1}. We call the word $\epsilon_2 \ldots \epsilon_{k+1}$ the *index-word* of the stack.

We define the notion of a *black* stack in the same way, by replacing all occurrences of *grey* by *black* in the just above definition.

At this point, consider a grey stack $F_1, \ldots, F_k, F_{k+1}$. Notice that if we wish to continue this stack, we have to develop F_{k+1}. There is a single outer sub-face of F_{k+1} which is adjacent to F_k. Let J_{k+1} be that face. Now we have to define the top of the new stack which is a sub-face of F_{k+1}. But four faces of F_{k+1} are adjacent to F_k and so, we have four possible tops for the stack. The number of the side which is common to J_{k+1} abd to the chosen top allows us to define the index-word of the new grey stack.

Let $F_1, \ldots, F_k, F_{k+1}$ be a grey stack and $G_1, \ldots, G_k, G_{k+1}$ be a black stack. We say that these stacks *fit in* together if F_1 and G_1 are adjacent and if G_1 is an inner sub-face of a face H_1 which is of the same generation as the face F_1 and if the indices of both stacks are complement: this means that if $\epsilon_1, \ldots, \epsilon_k$ is the index of the grey stack and if $\eta_1, \ldots, \eta_k$ is the index of the black one, $\eta_i = 3 - \epsilon_i$ for $1 \leq i \leq k$. Notice that this means that H_1 is a white face and that F_1 and H_1 are sub-faces of older faces F_0 and H_0, respectively, that F_0 and H_0 are white faces and that H_0 is processed before F_0: as long as H_1 is white and F_1 is grey, H_0 is a wall for F_0. The complementation of the indices of the stack says that in some sense the F_i's and G_i's are symmetric with respect to the common side of F_1 and G_1.

If $F_1, \ldots, F_k, F_{k+1}$ is a grey stack and if $G_1, \ldots, G_k, G_{k+1}$ is a black stack which fit in, we say that their *union* is a *face line* which goes from F_{k+1} to G_{k+1} or from G_{k+1} to F_{k+1}. We say also that F_{k+1} and G_{k+1} are the *ends* of the face line. We say that the edge which is common to F_1 and to G_1 is the *middle* of the face line. We say that k is the *length* of the face line.

We can now prove the following property:

Neighbouring lemma – *The ends of a face line are neighbours.*

Proof. Proceed by induction on the length of the stacks and then develop the ends of the face line for the induction step. Notice that the base step of the induction, when the length is 0, is contained in the Figures 2, 4 and 5 and constitutes the case of two neighbouring cells. The result is immediate from the definitions and by the generalization of the Figure 5.■

We have now to consider the reverse process. Starting from a white face, consider it as the end of some face-line. Is it possible to find the other elements of the face-line?

The answer is clear from the previous analysis. The considered white face is F_{k+1}, but we still don't know k. As F_k is not inside F_{k+1} it is one of the five neighbours of F_{k+1} and it is of the same generation as F_{k+1}. Consider the face F which contains F_{k+1} and which is of generation $gen(F_{k+1})-1$. If F_{k+1} is an outer face of F, then there is another face G with $gen(G) = gen(F)$ which is outside F and which shares with F the commons side of F and F_{k+1}. Then G is a white face and it is the other end of a face line of length 0 which consists of F_{k+1} and G. If a neighbouring outer face of F which is also a neighbouring face of F_{k+1} is grey, we can take it as F_k. There is now a face H with $gen(H) = gen(F)$, which is white and which has as a common side with G the side σ which is shared by F and F_k. If we develop H, the sub-face H_1 of H which shares σ is black. The index ϵ_{k+1} of F_{k+1} with respect to F_k is 0 or 3, and it is easy to see that if H_2 is the face of index $3-\epsilon_{k+1}$ with respect to H_1, H_2 is a white face and so, it is the end of a face line of length 1 which consists of F_{k+1}, F_k, H_1 and H_2. If F_{k+1} is an inner face of F, at least one of its outer face which has a common side with F_{k+1} is grey or black. In that case, we have to consider that shadowed face as F_k, assuming that it is grey: if it is black, we make a symmetric argument. Now we consider the face G_k which contains F_k and F_{k+1}. It is a white face, and as $gen(G_k) = gen(F_{k+1})-1$, we may proceed now by induction starting from G_k and we obtain that it is the end of a shorter line with F_{k+1} as an end. And so, it is easy to obtain an iterative algorithm from these considerations.

In order to make it more clear, we introduce some additional vocabulary. Having in mind the tree structure which we already defined, say that the sub-faces of a face F are the sons of F and that F is their father. If G is a fixed sub-face of F, we call *brothers* of G those sons of F which are adjacent to G. The top sub-face of F as well as the inner sub-faces of F have five brothers. An outer sub-face H of F has only four brothers and its common sides with a brother are different from its outer side σ.

Assuming that σ is not on the border of $\mathcal{O}$, among the faces which lie outside F and which has σ as a side, there is a single one, K, such that $gen(K)$ is maximal. It is easy to see that $gen(K) \in \{gen(H)-1, gen(H), gen(H)+1\}$. Indeed, consider L, the brother of F which also contains σ and call it the *uncle* of H. As L is the brother of F, $gen(L) = gen(F) = gen(H)-1$. If L is not white, it contains no sub-face and so, $K = L$ and, accordingly, we have $gen(K) = gen(H)-1$ in this case. If L is white, it contains an outer sub-face M which also contains σ. Then $gen(M) = gen(H)$. As L and F are adjacent white faces of the same generation, they have the same father and one is proceeded before the other one. Accordingly, H and M cannot be both white, one is grey and the other is white. If M is grey, then $K = M$ and $gen(K) = gen(H)$. If M is white, M contains a sub-face N which also contains σ. Now, as H is grey, N is black and it contains no sub-face. Accordingly $K = N$ and so $gen(K) = gen(H)+1$.

In the case when the outer side is on the border, H has no uncle and so, there is no face K outside the octant $\mathcal{O}$ which would contain σ.

Let us go back to the case when H has an uncle L. Of course H is the *nephew* of L. In the case when L is white, let K be its outer sub-face which contains σ. We say that K is a *cousin* of H and, conversely, that H is a cousin of K.

```
line(F, index_word) is
    -- initialization part
    liste := {F};
    if F ≠ outer_brother(F)
      then F_1 := outer_brother(F);
              liste := liste + {F_1};
              epsilon := index(F, F_1);
              F := F_1;
      else -- case when no shadowed outer brother:
              epsilon := ∅;
    end if;
    -- loop until the middle of the face line
    while colour(uncle(F)) ≠ white  -- assume never black
    loop liste := liste + {uncle(F)};
           epsilon := epsilon + {index(F,uncle(F))};
           F := uncle(F);  count := count + 1;
    end loop;
    -- initialization of the symmetric part
    index_word := epsilon; -- second step;
    if epsilon ≠ ∅
      then F := outer(uncle(last(liste)),last(epsilon));
              liste := liste + {F};
    end if;
    -- loop until a brother of the other end
    for i in 1 .. count
    loop pop(epsilon);
           F_1 := nephew(F, 3−last(epsilon));
           liste := liste + {F_1}; F := F_1;
    end loop;
    -- completion with the other end
    liste := liste + {brother(F, 3−last(epsilon))};
    return liste;
end line;
```

We see that the algorithm makes use of several auxiliary functions which we have now to define and which we shall consider as given elementary operations:

- *outer_brother*(F): returns a brother of F which is an outer sub-face of the father of F and which is grey or black, if any; if none, returns F;
- *index*(F, G): returns the index of F with respect to G; it is assumed that G is an outer face of its father and the index is taken with respect the outer side of G;
- *coulour*(F): returns the coulour of F: white, grey, black or dark;
- *uncle*(F): returns the brother of the father of F which shares a side with F; F is assumed to be an outer face;
- *outer*(F, i): returns the outer sub-face of the father of F which shares with F the side of index i, the basis of the indexes being the outer side of the outer sub-face;
- *last*(ℓ): polymorphic function being applied to any kind of list ℓ: returns the last element of ℓ, *i.e.* the element which is opposite to the top of ℓ;
- *nephew*(F, i): F being supposed to be an outer face of its father G, *nephew*(F, i) returns a sub-face G such that F is the uncle of G and such that G shares with F the side with index i taken from the outer side of F;
- *brother*(F, i): returns the sub-face of the father of F which shares with F the side of index i, the basis of the indexes being the outer side of the outer sub-face; the returned sub-face need not be outer;

These algorithms are not difficult to implement. It is enough to be able to define the sub-faces of a face as well as its sides and vertices. Also it is assumed that obtaining the father of a face is an elementary operation. With these primitives, it is not difficult to implement the above functions.

Rules of the neighbours:

The neighbours of a dodecahedron $\mathcal{D}_F$ *with the face* F *as its bottom are* *given as follows:*

- *the father of F, which is the face of the previous generation which contains F;*
- *the sons of F, which are the dodecahedra raised on the white faces of F;*
- *for each dark face of F if any, the node of another tree which is at the same place as this face;*
- *for each grey (black) sub-face of F, the other end of any face line which starts from F, being considered as a white face.*

2.4.2 Recovering the dual graph

Now, we turn to our last task in this geometrical part of the report: how to restore the dual graph from the spanning tree.

Indeed, the solution of this problem is a simple corollary of the previous algorithm: the sons of a face give access to one part of the neighbours. They correspond to the neighbours which are directly indicated by the spanning tree. The other connections are given by the face-lines which originate from the considered face. There are as many of them as there are shadowed faces.

Notice that in the plane, a rectangular grid is characterised by the property that the dual graph is constituted of elementary cycles with four nodes. In the $3D$ space, the rectangularity of the grid entails that precisely eight nodes enter the graph of vertices and edges of a cube. So, given a node, what are the seven other ones? Figure 8 gives an answer to the question in the simplest case: the first node is the face which contains the three ones which are displayed and which meet on P; next come the three faces themselves; next come the three white faces of the second generation which meet on P; the last node is the single white face of the third generation which shares P. The justification relies on the argument given with Figures 2, 4 and 5.

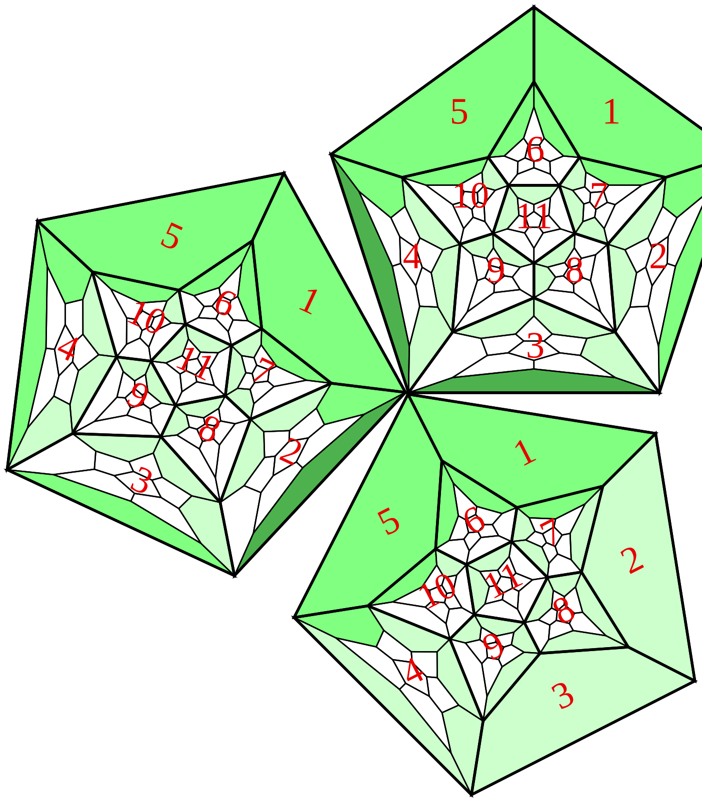


Figure 8. The situation of a corner

This situation can be generalised, already by considering Figure 8.

Indeed, call the 'big' faces of Figure 8 A, B and C, starting from the leftmost one and goning on clockwise. Consider the sub-faces 6 and 7 of A. These two faces meet on a point which also belongs to a grey face. The dodecahedron which has this grey face as a subface exists: it is the dodecahedron being inscribed in the sub-face 4 of B, say 4_B. And so, the bottom of the 'third' dodecaedron, which is also a neighbour of A is the face 4_B, see Figure 9. With these three sub-faces which meet on a point Q, we have the three dodecahedra of the first generation. For the second generation, the sub-faces 6 and 7 of A yield a single white sub-face which shares the point Q. When we develop the face 4_B, the point Q appears as the vertex which is shared by the black sub-face of the face 4_B and two white sub-faces of the face 4_B. These two white sub-faces also belong to the expected second generation. When we develop these sub-faces to their turn, The area around Q

in the face 4_B is now splitted into four pentagons. It is easy to see that two of them are black because they are adjacent to the black sub-face of 4_B. On the two remaining which are adjacent sub-faces of two adjacent white faces, one is grey and the other is white: it yields the single dodecahedron of the third generation and we are done.

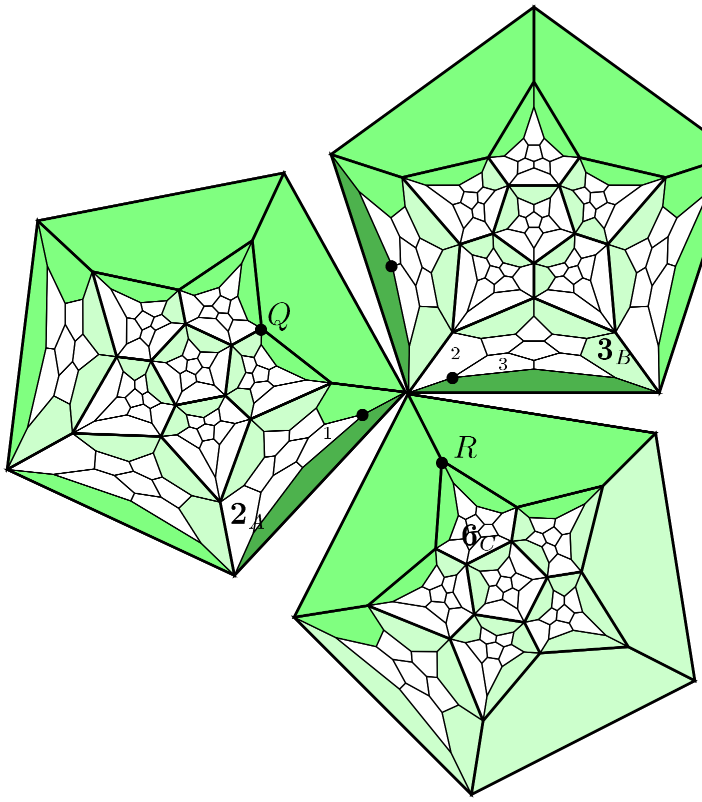


Figure 9. Other situations of a corner

As a last example, consider the point R. It is in the face C at the intersection of to grey faces, the faces 5 and 1, and a white one: the face 6. The dodecahedra which are outside C and which share its grey faces are based respectively on the faces 2_A and 3_B. And so, the faces 6_C, 2_A and 3_B are the three dodecahedra of the first generation for the considered cube. Next, we see that in C, the point R is completely surrounded by grey faces. Fortunately, the point appears in two other places of the picture, in the face 2_A and in the face 3_B, see Figure 9. Around these points, which also coincide with R, we see three white faces of the second generation which are labelled

1, 2 and 3 in Figure 9. When these faces are developped, it is easy to see that the point R in the face being labelled 1 is completely surrounded by grey faces. For the point R which is in the face 3_B, the faces 2 and 3 split into four sub-faces around R. Two are black as long as they are adjacent to black faces of B, and for the remaining two sub-faces, one is grey and the other is white, which provides us with the single white face of the third generation.

Now, we can see that these situations can be generalised as we generalised the situation of Figure 4 to face lines. In the present case, the point of intersection P of the three faces of the first generation can be split into two or three copies which we find thanks to the process which is described by the face line algorithm. The situation of Q or R is repeated.

In the case of a splitting into two points, we find one face of the second generation in the point which is common to two faces of the first generation and we find two faces of the second generation in the copy which belongs to a single white face of the first generation.

In the case of a splitting into three points, one is shared by a single white face of the first generation, each one of the two other copies of the point belong to one of the two other white faces of the first generation. The first copy is surrounded by grey faces of the next generation. For the two remaining points: one is shared by a single white face of the second generation and so, it is surrounded by shadowed faces at the level of the third generation; the other one is shared by two white faces of the second generation and a single white sub-face survives at the level of the third generation.

With this last unformalized algorithm, we completed the geometric part of the report and we turn now to the numerization of what was previously described.

3 The numbering

Recall that Figure 6 indicates the decomposition of any simple set as we defined them in the previous section. From this figure, we already extracted rules to construct the spanning tree of the dual graph of $\mathcal{T}$. These rules give the table of the numbers of simple sets of each type which are produced by the splitting of a given simple set. We may display this table in the format of a 4×4 matrix M:

3:	5	0	3	1
2:	4	0	3	1
1:	3	0	3	1
0:	2	1	2	1

In section 2.4, we saw how we can construct a tree T from this information. We already noticed that this construction of T provides us with a *planar* tree. However, the splitting process which we described in great detail in section 2 gives rise to a spanning tree which in some sense the skeletton of the splitting and so, it is a tree in the $3D$ hyperbolic space. And so, one of our tasks will be to make more explicit the correspondance between T and the $3D$ spanning tree.

Recall that the root of T is a 3-node. The root has 9 sons: five ones are 3-nodes too, three ones are 1-nodes and a single one is a 0-node. For 2-nodes, 1-nodes and 0-nodes, we associate, respectively, the second line, the third one and the fourth one of the matrix in the same with which we used the first line to organise 3-nodes.

Repeating what was introduced in [20] for the pentagrid of $I\!H^2$ and which was generalised in [**?**, **?**] to any rectangular regular grid of $I\!H^2$, we shall here introduce a numbering of the nodes of T.

As we said in the introduction, we expect to extract the numbers of the neighbours of a node from its own number thanks to a suitable representation. This could be done by very simple algorithms in $I\!H^2$. Here, the difficulties which we encountered with the geometrical approach points at a significant difference. We shall see that this is the case.

Consider that we have numbered the nodes of T by positive numbers, starting from the root by 1 and then going on from level n to level $n{+}1$, from left to right on a same level.

We notice that the sum of the first line of M gives the number of nodes which are sons of the root and, by induction, it is easy to prove that the sum of the first line of M^n gives the number of the nodes which are on the n^{th} level of T.

Now, a routine computation shows that the minimal polynomial of M is

$$(X-1)(X^2-8X+1) \qquad (1)$$

From that, we know that if u_n is the number of nodes of level $n{-}1$, these numbers obey the following recurrence relation:

$$u_{n+3} = 9u_{n+2} - 9u_{n+1} + u_n \qquad (2)$$

with $u_{-1} = u_0 = 0$ and $u_1 = 1$ as initial conditions.

Now, denote by a_n, b_n, c_n and d_n the respective numbers of nodes of types 3, 2, 1 and 0 on the n^{th} level. These numbers also obey the recurrence equation (2) with, of course, other initial conditions. We shall keep these notations to denote also the respective numbers of nodes of types 3, 2, 1 and 0 on the n^{th} level of a *sub-tree* of T, rooted starting from a given node. These numbers depend only on the type of the node which is taken as a root of the sub-tree. We get easily that $u_n = a_n + b_n + c_n + d_n$ when the sub-tree is T itself. We denote the sum $a_n + b_n + c_n + d_n$ by v_n, w_n and t_n when the root of the sub-tree is, respectively, of types 2, 1 and 0.

3.1 The greedy representation of numbers.

First, we recall the main definition and properties of regular greedy representations, see [7, 11].

Let $\mathcal{U} = (u_n)_{n\geq 1}$ be a strictly increasing sequence of positive integers with

- $u_1 = 1$,
- $1 < D = \sup \dfrac{u_{n+1}}{u_n} < \infty$

An algorithmic enumeration system for the natural numbers can be constructed from this sequence. The algorithm is called *greedy* and it works as follows:

Let $b = max\{n|n < D, n \in N\}$. The set $A_{\mathcal{U}} = \{0, 1, \ldots, b\}$ is called the canonical alphabet being associated with the sequence $\mathcal{U}$. Its elements are called digits. Let $A^{\star}_{\mathcal{U}}$ be the set of all words written with the letters of $A_{\mathcal{U}}$.

For every $n \in N$ a word $U(n) = \alpha_k \ldots \alpha_1 \in A^{\star}_{\mathcal{U}}$ with

$$n = \alpha_k u_k + \cdots + \alpha_1 u_1, \quad \alpha_k \neq 0 \quad \text{for} \quad n \neq 0 \tag{3}$$

which is given by the following algorithm:

-- n is given
-- first, compute the size of $U(n)$:
$i := 1$;
while $u_i \leq n$ **loop** $i := i+1$; **end loop**;
-- now, $i = \min\{j \mid u_j > n\}$
-- as the size is now, compute the digits of $U(n)$, downward:
$U := \emptyset$;
for k **in** $1 \,..\, i-1$ **in reverse**

loop $U := U \;\&\; \{\lfloor \frac{n}{u_k} \rfloor\}$;
$\quad n := n \bmod u_k$;
end loop;

The word $U(n) = \alpha_k \dots \alpha_1 \in A^\star_{\mathcal{U}}$ is called the normalised *greedy representation* of the integer n with respect to the basis sequence $\mathcal{U}$. We shall write also

$$n \propto \alpha_k \dots \alpha_1$$

and

$$\overline{\alpha_k, \dots, \alpha_1} = \alpha_k u_k + \dots + \alpha_1 u_1.$$

Notice that, in general, the representation (3) is not unique. However, the word $U(n)$ which is defined above is uniquely determined. It boils down to introduce the lexicographical order in the set of all representations (3) of a given natural number n: the word $U(n)$ is the maximal one with respect to this order.

This is a consequence of the following property:

Say that the word $\alpha_k \dots \alpha_1$ is a normalised greedy representation if it is the normalised greedy representation of some positive number with respect to the basis sequence $\mathcal{U}$. Then:

Proposition: – *Let*

$$n = \alpha_k u_k + \dots + \alpha_1 u_1, \alpha_j \in A_{\mathcal{U}}, \quad \alpha_k \neq 0.$$

Then the word $\alpha_k \dots \alpha_1$ is a normalised greedy representation iff

$$\alpha_i u_i + \dots + \alpha_1 u_1 < u_{i+1}, \text{ for } 1 \leq i \leq k.$$

Proof. By induction on the length of $\alpha_k \dots \alpha_1$. ■

Another important property is the following:

Comparison lemma – *Let $\alpha_k \dots \alpha_1, \beta_k \dots \beta_1 \in A^\star_{\mathcal{U}}$ and $\alpha_i \leq \beta_i$, for all i in $1, \dots, k$, with $\alpha_k \neq 0$. Then*

- *If $\beta_k \dots \beta_1$ is a normalised greedy representation, then $\alpha_k \dots \alpha_1$ is also a normalised greedy representation,*
- *if $\alpha_k \dots \alpha_1$ is not a normalised greedy representation,then $\beta_k \dots \beta_1$ is not a normalised greedy representation.*

Proof. It is enough to prove the first assertion. Assuming that $\alpha_i \leq \beta_i$, for all i in $1, \ldots, k$, apply the greedy algorithm to the number $m = \sum_{i=1}^{k} \alpha_i.u_i$. From the previous proposition, the size of m is also k. Then, from the same proposition and as $\alpha_i \geq 0$, we have that $\lfloor \frac{n}{u_k} \rfloor$ is α_k. By induction on k, it is easy to see that this property is kept in the loop of the greedy algorithm.■

Examples

Here are a few illustrative examples. The first one deals with the classical reresentation of numbers. The second one is already less standard and makes use of the Fibonacci sequence which was already used in [14, 20] for the regular rectangular pentagonal tiling of $I\!H^2$. The last example concerns the family of greedy representations which we use in this paper.

1. Let $b \geq 2$ be a natural number. Let $\mathcal{U} = (b^n)_{n \geq 0}$. The canonical alpphabet associated with $\mathcal{U}$ (the set of digits) is $A_{\mathcal{U}} = \{0, 1, \ldots, b-1\}$. Then the normalised representation of the natural number n with respect to sequence $\mathcal{U}$ is the b-adic (radix) representation of n in the basis b.

2. Let $\mathcal{F} = (F_n)_{n \geq 1}$ be the sequence of the Fibonacci numbers. The canonical alphabet is $A_{\mathcal{F}} = \{0, 1\}$. A word $\alpha_k \ldots \alpha_1 \in A_{\mathcal{F}}^{\star}$ is a normalised representation iff $\alpha_k \neq 0$ and the block 11 is not a subword of $\alpha_k \ldots \alpha_1$, see [7].

3. Let $s \geq 5$ be a natural number. Consider the greedy numeration system with respect to the basis sequence $\mathcal{U}_s = (u_n)_{n \geq 1}$ which satisfies the recurrence relation

$$x_{n+2} = (s-2)x_{n+1} - x_n \tag{4}$$

with initial conditions $u_1 = 1, u_2 = s - 2$.

The characteristic equation of this recurrence is

$$P(x) = x^2 - (s-2)x + 1 = 0.$$

Its roots are $\beta_s > 1$ and β_s^{-1}.

The root β_s is dominant: it is a Pisot number, *i.e.* $0 < \beta_s^{-1} < \beta_s$.

Moreover

$$u_n = \frac{1}{\sqrt{s(s-4)}}(\beta_s^{n+1} - \beta_s^{-n-1})$$

and $\beta_s = \lim_n \frac{u_{n+1}}{u_n}$.

Notice also that $P(x) < 0$ when $x \in]\beta_s^{-1}, \beta_s[$, that $P(s-2) = 1$ and that $P(s-3) < 0$. Consequently, we obtain that

$$s - 3 = max\{n | n < \beta_s, n \in N\}, \tag{5}$$

which means that the canonical alphabet is $\mathcal{U}_s = \{0, \ldots, s-3\}$.

This example satisfies the requirements of theorem 8.1 from [11]. Accordingly, the set of the normalised greedy representations of the natural numbers is a *regular* language. This language corresponds to the numbering of the nodes of the tree which is associated, in [**?**, **?**], to any rectangular grid of the hyperbolic plane. In [**?**, **?**], there is also a simple proof of this property for the languages which is defined in Example 3, which directly follows from the construction of the tree.

3.2 The language associated to the greedy representation of numbers induced by T.

And so, considering the sequence $\{u_n\}_{n \geq 1}$ which we defined by the recurrence equation (2) and the corresponding initial conditions, we have that any positive number n can be represented in the form $n = \sum_{i=1}^{k} a_k u_k$, with $a_k \in \{0..8\}$. Indeed, notice that the minimal polynomial of M has $X^2 - 8X + 1$ as a factor. This polynomial is associated to the equation (4) for $s = 10$. The estimate (5) gives us the condition $a_k \in \{0..8\}$.

We obtain that the growth of $\{u_n\}_{n \geq 1}$ is exponential.

From now on, we shall consider the normalised greedy representation of natural numbers in the basis $\{u_n\}_{n \geq 1}$. In order to make our text as lighter as possible, we shall say *greedy representation* for normalised greedy representation in the basis $\{u_n\}_{n \geq 1}$. We shall also denote the greedy representation of n by n_u. Recall that if the word ρ is the greedy representation of a number, $\overline{\rho}$ is the number which is represented by ρ.

Theorem 2 (Margenstern) – *The language of the normalised greedy representation in basis $\{u_n\}_{n \geq 1}$ of the positive numbers is not regular, it is even not context-free.*

Before proving the theorem, we need several technical lemmas in order to establish the following property.

Below, Table 1 displays the first terms of the greedy representation of the numbers $\{u_n - 1\}_{n \geq 2}$.

The table suggests a recurring pattern which we shall now analyse in the greedy representation of $u_n - 1$. It is the content of Lemmas 1 and 2 which we shall first prove. To that purpose, we introduce some additional notation. Define $||n||_u$ to be the number of digits of the number n. If $n \propto \alpha_1, \ldots, \alpha_k$, the number of digits is k, which we write also as follows: $||\alpha_1, \ldots, \alpha_k||_u = k$. Also, denote $||n||_\alpha$ to be the number of α's that occur in the greedy representation of n.

1	8	6	766673	11	76666666678
2	78	7	7666674	12	766666666700
3	770	8	76666675	13	7666666666701
4	7671	9	76666676	14	7666666666702
5	76672	10	7666666677		

Table 1. Greedy representation of the first $u_k - 1$.

Lemma 1 – *For all $k \geq 4$, we have:*

$$u_k - 1 =_u 76^{k-\ell(k)} 7\rho_k \tag{6}$$

where ρ_k is the greedy representation of a positive number. We have that $\overline{\rho_k}$ obeys the following induction:

$\overline{7\rho_{k+1}} = \overline{7\rho_k} + 1$, except when $\overline{7\rho_k} + 1 = u_k$;
when $\overline{7\rho_k} + 1 = u_k$, $\overline{\rho_{k+1}} = 0$ and $||\rho_{k+1}||_u = ||\rho_k||_u + 1$.

Before proving the lemma, call *time of resetting* the value of k when $\overline{7\rho_k} = u_k - 1$. This notion allows to characterise ρ_k more precisely:

Lemma 2 – *Denote by $\{r_i\}_{i \geq 1}$ the times of resetting of ρ_k defined in lemma* 1. *Then:*

$r_{i+1} = r_i + u_i - u_{i-1} + 1$ for $i \geq 2$;

And so, between two consecutive times of resetting, the value of ρ_k increases from 0 until $u_h - 7u_{h-1} - 1$, where $h = ||\rho_k||_u$.

Proof of lemma 1: The proof relies on a technical lemma:

Sub-lemma – *For all $k \geq 2$, $-6u_k + 6.9u_{k-1} - 6u_{k-1} - 6u_{k-2} = -6$.*

Proof of the sub-lemma: induction on k. For $k = 2$, $6u_k - 6.9u_{k-1} +$

$6u_{k-1} + 6u_{k-2} = 6.9 - 6.9.1 + 6.1 + 6.0 = 6$.

For $k \geq 2$, $6u_{k+1} - 6.9u_k + 6u_k + 6u_{k-1} = -6.9u_{k-1} + 6.u_{k-2} + 6u_k + 6u_{k-1}$ by using (2) multiplied by 6. From that, $6u_{k+1} - 6.9u_k + 6u_k + 6u_{k-1} = 6.u_k - 6.9u_{k-1} + 6u_{k-1} + 6.u_{k-2}$. ■

Let us now turn to the proof of lemma 1.

We argue by induction on k. Assume that for all $h \leq k+3$,

$$u_{k+3} - 1 = 7u_{k+2} + 6\sum_{i=h+2}^{k+1} u_i + 7u_{h+1} + \sum_{i=1}^{h} a_i u_i$$

From the recurrence relation (2), we get:

$$\begin{aligned}
u_{k+4} - 1 = \;& 9\Big(7u_{k+2} + 6\sum_{i=h+2}^{k+1} u_i + 7u_{h+1} + \sum_{i=1}^{h} a_i u_i\Big) \\
& - 9\Big(7u_{k+1} + 6\sum_{i=h+2}^{k} u_i + 7u_{h+1} + \sum_{i=1}^{h} a'_i u_i\Big) \\
& + 7u_k + 6\sum_{i=h+2}^{k-1} u_i + 7u_{h+1} + \sum_{i=1}^{h} a"_i u_i \\
= \;& 7(9u_{k+2} - 9u_{k+1} + u_k) + 6\sum_{i=h+2}^{k-1} (9u_{i+2} - 9u_{i+1} + u_i) \\
& + 9.7u_{h+1} - 9.7u_{h+1} + 7u_{h+1} + \sum_{i=1}^{h} (9a_i u_i - 9a' u_i + a"_i u_i) \\
& + 6.9u_{h+3} \\
= \;& 7u_{k+3} + 6\sum_{i=h+5}^{k+2} u_i + \; 6.9u_{h+3} + \; 7u_{h+1} \\
& + \sum_{i=1}^{h} (9a_i u_i - 9a' u_i + a"_i u_i)
\end{aligned}$$

Now, by (2), $6.9.u_{m+2} = 6.u_{m+3} + 6.9.u_{m+1} - 6.u_m$, and so,

$$\begin{aligned}
u_{k+4} - 1 = \;& 7u_{k+3} + 6\sum_{i=h+4}^{k+2} u_i + \; 6.9u_{h+2} - 6.u_{h+1} + \; 7u_{h+1} \\
& + \sum_{i=1}^{h} (9a_i u_i - 9a' u_i + a"_i u_i) \\
= \;& 7u_{k+3} + 6\sum_{i=h+4}^{k+2} u_i + \; 6.9u_{h+2} - 6.u_{h+1} + \; 7u_{h+1} + C_h
\end{aligned}$$

By the sub-lemma, $6.9.u_{h+2} - 6.u_{h+1} = 6u_{h+3} + 6u_{h+2} - 6$a. And so,

$$u_{k+4}-1 = \ 7u_{k+3} + 6\sum_{i=h+2}^{k+2} u_i + \ 7u_{h+1} + C_h - 6$$

Now, by induction hypothesis, $\overline{7\rho_{k+3}} = \overline{7\rho_{k+2}} + 1 = \overline{7\rho_{k+1}} + 2$. Accordingly,

$$\begin{aligned}
7u_{h+1} + C_h &= 7u_{h+1} + \sum_{i=1}^{h} (9a_iu_i - 9a'u_i + a"_iu_i) \\
&= 9.7u_{h+1} - 9.7u_{h+1} + 7u_h + 1 + \sum_{i=1}^{h} (9a_iu_i - 9a'u_i + a"_iu_i) \\
&= 9\overline{\rho_{k+3}} - 9\overline{\rho_{k+2}} + \overline{\rho_{k+1}} = \overline{\rho_{k+1}} + 9.
\end{aligned}$$

And so, if we put $c_i = 9a_i - 9a'_i + a"_i$, we get:

$$u_{k+4}-1 = \ 7u_{k+3} + 6\sum_{i=h+2}^{k+2} u_i + \ 7u_{h+1} + C_h - 6,$$

and, from the previous computation, $7u_{h+1} + C_h - 6 = \overline{\rho_{k+1}} + 3$, which proves that $\overline{\rho_{k+4}} = \overline{\rho_{k+3}} + 1$.

To complete the proof of the lemma, we have to see what happens when k reaches a resetting time and the two steps after, because our recurrence deals with three terms of the sequence.

Assume now that $\overline{7\rho_{k+3}} = u_{h+2}-1$. By our hypothesis of reccurence which was valid for k+3, k+2 and k+1, we conclude that the same computation is valid and we obtain, in particular, that $\overline{7\rho_{k+2}} = u_{h+2}-2$ and $\overline{7\rho_{k+1}} = u_{h+2}-3$. Accordingly, in that case,

$$\begin{aligned}
u_{k+4}-1 &= \ 7u_{k+3} + 6\sum_{i=h+2}^{k+2} u_i + \ 7u_{h+1} + C_h - 6, \\
&= \ 7u_{k+3} + 6\sum_{i=h+2}^{k+2} u_i + 9\overline{u_{h+2} - 1} - 9\overline{u_{h+2} - 2} + \overline{u_{h+2} - 3} - 6 \\
&= \ 7u_{k+3} + 6\sum_{i=h+2}^{k+2} u_i + u_{h+2}. \\
&= \ 7u_{k+3} + 6\sum_{i=h+3}^{k+2} u_i + 7u_{h+2}.
\end{aligned}$$

which is the form of the resetting time. If we assume now that $\overline{1\rho_{k+3}} = u_{h+2}$ and, consequently, that $\overline{7\rho_{k+2}} = u_{h+2} - 1$ and $\overline{7\rho_{k+1}} = u_{h+2} - 2$, taking into account that $7u_{h+2} = 6.u_{h+2} + u_{h+2}$, we get that this time, $7u_{h+1} + C_h - 6 = 9u_{h+2} - 9(u_{h+2} - 1) + u_{h+2} - 2 - 6 = u_{h+2} + 1$. This allows us to perform a new step of the computation, assuming now that $\overline{1\rho_{k+3}} = u_{h+2} + 1$, that

$\overline{7\rho_{k+2}} = u_{h+2}$ and $\overline{7\rho_{k+1}} = u_{h+2} - 1$, now we have $7u_{h+1} + C_h - 6 = 9(u_{h+2} + 1) - 9u_{h+2} + u_{h+2} - 1 - 6 = u_{h+2} + 2$. And now, we may start again the induction until the next resetting time.

At last, let us look once again at what happens in the resetting time. This can be illustrated as follows:

7	6…66	7	6 α_{h-1} … α_1	$u_k - 1$
76	6…66	0	00 0	$u_k - 1$
	1	0	00 0	
76	6…67	0	00 0	$u_{k+1} - 1$

This way, the transformation of $u_{h+2}-1$ compensate the appearence of a new digit 6 in the first block of 6's: the leading 1 of u_{h+2} is added to the last 6 of this block, and so the umber of leading 6's remains the same. Nevertheless, the equality $||u_{k+1}||_u = ||u_k||_u + 1$ is sitll observed.■

The proof of Lemma 2 is straightforward from the proof of Lemma 1.■

Proof ot Thoerem 2:

Denote the language of the greedy representations of the natural numbers by $\mathcal{L}$. Consider the subset constituted by the following infinite subsequence of the greedy representations of u_k-1:

$$\mathcal{M} = \{76^{s(k)}70^k \ ; \ \exists i \ \overline{76^{s(k)}70^k} = u_i - 1\},$$

where, according to Lemma 2, we know that $s(k)$ is an increasing fonction with an exponential growth in k.

Notice that $\mathcal{M} = \mathcal{L} \cap \{76^*70^*\}$

Consequently, if $\mathcal{L}$ were context-free, so were $\mathcal{M}$. It would be possible to apply the pumping lemma for context free languages to $\mathcal{M}$. As $\mathcal{M}$ is infinite, we could find one of its elements, say for some fixed k, such that $76^{s(k)}70^k = xuyvz$ with uv non empty and such that for all h, xu^hyv^hz would be in $\mathcal{M}$. Necessarily, $uv \in 6^+$ or $uv \in 0^+$ or both $u \in 6^+$ and $v \in 0^+$. It is immediate that these three possibilities are in contradiction with the exponential growth of $s(k)$ and with its injectivity.■

Remark (G. Skordev)

Here, we give an alternative proof that the language $\mathcal{L}$ of the normalised greedy representation of the natural numbers with respect to the basis $\{u_n\}_{n\geq 1}$ is not regular.

The sequence $\{u_n\}_{n\geq 1}$ satisfies the recursion:

$$x_{n+3} = 9x_{n+2} - 9x_{n+1} + x_n$$

The characteristic polynomial of this recursion is:

$$P(X) = X^3 - 9X^2 + 9X - 1 = (X-1)(X^2 - 8X + 1).$$

The recursion is the minimal one for $\{u_n\}_{n\geq 1}$. Therefore, if $\{u_n\}_{n\geq 1}$ satisfies some other recursion with charasteristic polynomial $Q(X)$, it follows that $P(X)$ divides $Q(X)$.

The root of $P(X)$ are β, β^{-1} and 1, with $\beta = 4 + \sqrt{15}$, the dominant root. The β-representation of 1 is:

$$1 = \frac{7}{\beta} + \sum_{i=2}^{\infty} \frac{6}{\beta^i}.$$

It has a minimal lead one and a minimal period one. Assume that the language $\mathcal{L}$ is regular. Then, from Theorem 8.1 of [11], it follows that $\{u_n\}_{n\geq 1}$ satisfies a recursion with characteristic polynomial

$$b(X) = X^{\ell-1}(X^{p-1} + X^{p-2} + \ldots + X + 1)(X^2 - 8X + 1).$$

Accordingly $P(X)$ should divide $b(X)$. This is impossible.

4 The numerical algorithms

The result of this section belong to the first author.

In the section, we consider how to 'numerise' the geometrical algorithm which we developed in section 2 in order to find the neighbours of a cell.

Having in mind what was introduced and performed in [20], which was continued in [22, 23], we shall try to devise an algorithm such that, given a cell by its number, find the numbers of its neighbours.

In order to facilitate the task, we shall first assume that the neighbours of the cell are all in $\mathcal{O}$. As a preliminary step, we shall study a connected problem: find the path in T from a node to the root, again from the number of the node.

4.1 The numbering path and the local maps.

As we noticed in section 2, the rules (R) of the sub-section 2.4 define a planar tree which we denoted T. However, the splitting process which is described in the same section is a spatial process. It defines a tree in $I\!H^3$, let us denote it $\mathcal{A}$, and our first goal will consists in establishing a bijection between T and $\mathcal{A}$.

As T is planar, there is a natural order on T which is given by the numbering of the nodes, level after level and, on each level, from the leftmost node to the rightmost one. In the sub-section 2.3, we noticed that the splitting process induces an order on the white sub-faces of a face. We noticed also that in the case of an octant, there are basically two *natural* ordering of the sub-faces which are induced by the splitting process, the direct order and the reverse order. Below, Figure 10 represents the direct order of the octant. We materialise this oder as a *path* which connects the white sub-faces, one after another. We call this path the *numbering path*.

Of course, there are other numbering paths: two for each kind of simple sets which we defined in the splitting process of the sub-section 2.3. See Figure 12 for a representation of the numbering paths in the direct order for the half-octant, the corridor and the tunnel.

Below, in Figure 11, we make one step further in the case of the octant. It is now not difficult to notice that the path of the second generation can be generated by using the maps of Figure 12 which, for this reason, are called *local maps*. We see that compared with the map of Figure 10, there is an additional information. We put in yellow the faces for which the ordering of the next generation is the reverse order. It can be noticed that indeed, the path of second generation for the octant can be obtained from the maps of Figure 12 and their symmetric counterpart with the reverse order, which do not appear in Figure 12.

Notice that we need still another information: in the local maps, the end point of the path is always direct. This indeed raises no problem, as long as the end point is always a tunnel: in that case, there is *a priori* no meaningful difference between the reverse or the direct order.

Octants have exactly two shadowed faces. There is a single inner face which is adjacent to both of them: by defintion, this face is the face 6. Each face of a Schlegel diagram is oriented by the planar orientation of the whole diagram and we consider the clockwise orientation in order to fix the numbering of the sub-faces of any face. With these conventions, the numbering path within an octant always start from the sub-face 6 and it ends in the sub-face 4 in the case of a direct order, by the sub-face 2 in the case of a reverse order.

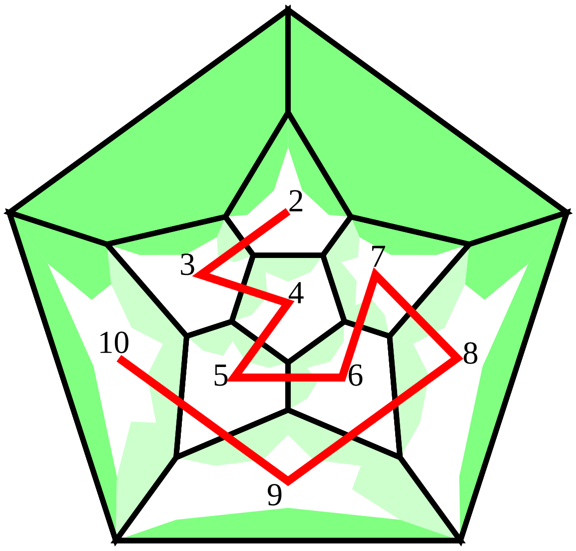


Figure 10. The numbering path: the octant, first generation.

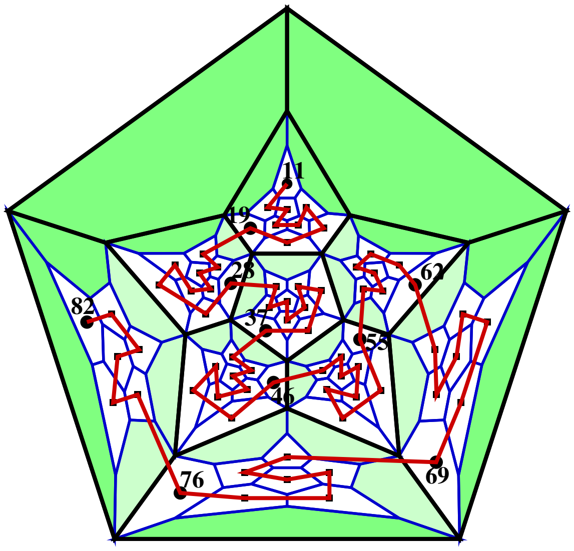


Figure 11. The numbering path: the octant, second generation.

Half-octants have three shodowed faces. By definition, its white outer sub-faces are the sub-faces 3 and 4, and thanks to the orientation, this fixes the number of all sub-faces. Again, the numbering starts from the sub-face 6 and it ends in the sub-face 4 when the order is direct, in the sub-face 3 when we have the reverse order.

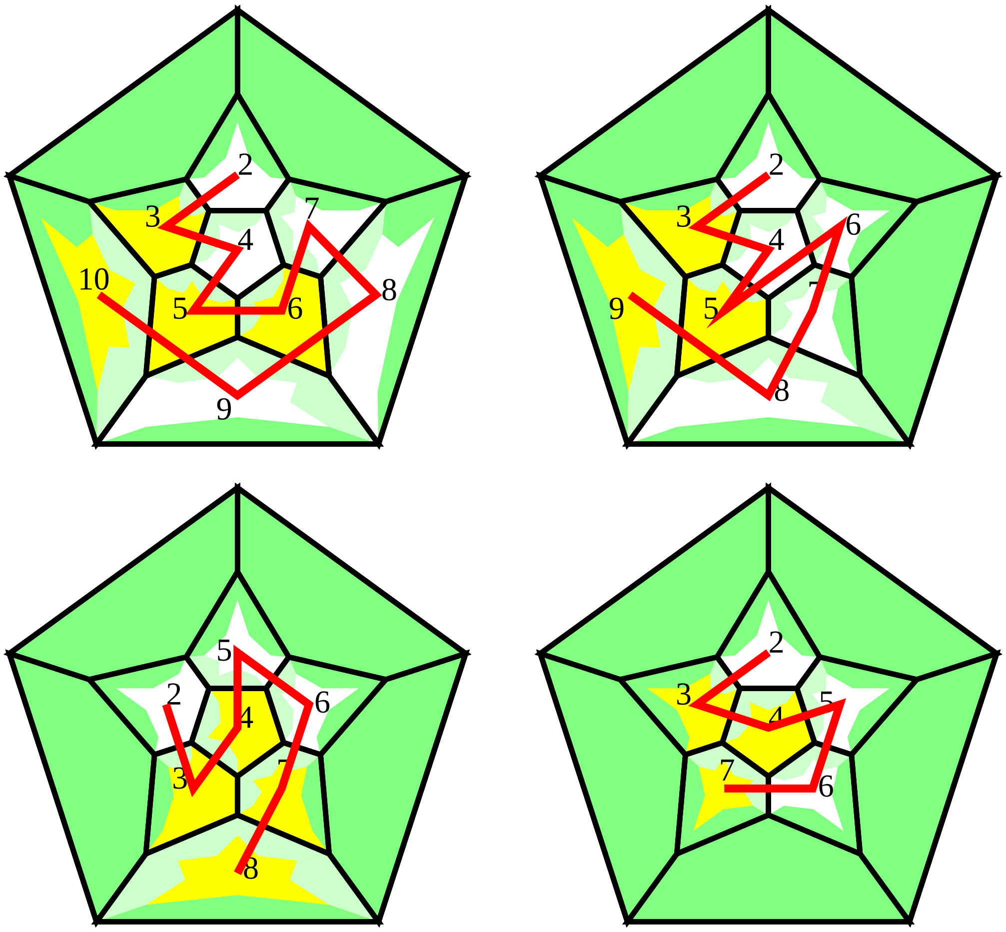


Figure 12. Construction of the numbering path:
The local maps with the direct order for the octant, the half-octant, the corridor and the tunnel.
The yellow faces indicate a use of the reverse order for the next generation.

Corridors have a single white outer face. By definition, it is the sub-face 3 which, again thanks to the orientation, fixes all other numbers. Here, the numbering path starts from the sub-face 10 in the case of a direct order, from the sub-face 7 in the case of a reverse order, and it always end in the face 3.

For tunnels, the situation is more complex. In fact, the starting face is fixed by the ending sub-face of the corridor by which the path arrives. The sub-face 3 of that corridor is adjacent to a grey face of the tunnel which, by definition will be the sub-face 1. Thanks to the oriention, this fixes the numbers of all the other sub-faces of the tunnel. The numbering path is reverse, in order to have the ending point adjacent to an outer face which is turned to the corresponding brother. The path starts from the sub-face 6 and it ends in the sub-face 9.

local number	global number	local number	global number
1	(2,7)	7	(7)
2	(8)	8	(6)
3	(9)	9	(5)
4	(10)	10	(3)
5	(2,3)	11	(4)
6	(2)		

Table 2. Correspondance between local and global coordinates of faces.

We may also number the sub-faces by the numbers which are defined by the numbering path. In this way, the white faces are defined by their number which is induced by the numbering path. For grey faces, they are numbered with respect to their white inner neighbours. As an example, in the first octant, where the white sub-faces are numbered from 2 up to 10, Table 2 gives us the correspondance between the previous labelling and the one which we now consider.

As an other example, consider the dodecahdron $\mathcal{D}_7$ whose white subfaces are numbered by the path from 56 up to 62. Table 2 becomes now the following Table 3:

Notice that we obtain the table for $\mathcal{D}_9$, for instance, simply by advancing the above global numbers by 14. This is a general feature: local maps of faces of the same type and same orientation can be computed from one another by simple shifts. The shift is the *distance*, along the numbering path, between the two starting points of the considered faces.

Next, as the *numbering path* defines an *order* between the faces, *i.e.* a face which receives a smaller number is processed *before* a face which receives a bigger one. Consider Figure 1 which is one of the local maps. If we know the kind of shadow of faces (2,3) and (2,7), we know the kind of shadows for the sub-faces of 2, 3, 4, 5, 6 and 7. If we know what is beyond faces 8, 9 and 10, *i.e.* on the other side of their respective outer side, we know also the of shadows of the subfaces of faces 8, 9 and 10. And, of course, thanks to the local map that is associated to the father of the node that we consider, we know the corresponding information. In case we are in the local map of the root, we also know.

local number	global number	local number	global number
1	(59,60)	7	(60)
2	(60,61)	8	(61)
3	(62)	9	(57)
4	(56,57)	10	(56)
5	(56,59)	11	(58)
6	(59)		

Table 3. Another correspondance between local and global coordinates of faces.

And so, in all cases, the local maps allows us to know also the kind of shadow that a shadowed map receives.

4.2 The numerical algorithms

In this section, we construct two algorithms: the first one gives the path from a cell to the root; the second one gives the neighbours of a cell. In both cases, the input is the number of the cell. For the path, the output is a list of numbers: the numbers of the nodes which lie on the path from the root to the cell. For the neighbours, the algorithm gives a list of twelve numbers which are the numbers of the neighbours of the cell.

4.2.1 The algorithm for the path

We start with an example which illustrates how the algorithm works.

We consider the node with the number $n = 712674_u$.

First, we determine which level the node belongs to. We now that on each level k, there is a change of length by the occurrence of u_k. In order to see whether a node with $k{+}1$ digits belongs to the level k or to the level $k{+}1$, we compare it, lexicographically, with 1^{k+1} which is the leftmost node of the level k. If n is greater than this bound, it belongs to the level k, otherwise, it belongs to the level $k{+}1$. In the present case, $1^6 < n < 1^7$ and so, n belongs to the level 6.

In order to compute the path from n to the root, we start from the root, and we determine, level after level, to which sub-tree being rooted on the considered level the node does belong.

And so, for each node m of the first level, we compute the righmost node of the sub-tree which is rooted in m. We start from the leftmost border of T and, adding the width of the sub-tree being rooted in j for the considered level and starting with $j = 1$, we increment j until the bound is at least equal to n. The value of j for which n is reached or bypassed gives us the son m of the root such that the sub-tree rooted in m contains n. During the computation, each time when j is incremented, we update the values of the 'border' which now is the rightmost branch of the sub-tree with m as its roots.

Now, this process is repeated with the next level for the sons of m. This will give us a new sub-tree, being rooted in a son of m which contains n. We proceed in the same way and we update the border when the expected node is found.

The process is repeated until we are at the level of n.

Consider that we are on a node ν of type s. We say that the number of nodes of the tree T_ν being rooted in ν at the level k is $w(k,s)$. Consider now the above process and let $b(k)$ the position of the border at the level k of T. Assume that we know that n is in the sub-tree which is rooted in m, where m is a node of the level h in T. We know also the type of m, say s, and so, we first add $w(k{-}h,1)$ as long as needed and as no more times than the number of 3-nodes among the sons of m. If n is not yet found, then we arrive to the sons of type 2. Then we add $w(k{-}h,2)$, and so on. In order to uniformise this situation, we introduce a function $\delta(k,s,i)$ such that:

$$\delta(k,s,i) = \begin{cases} w(k,3) & \text{if } 1 \leq i \leq M_{1,s} \\ w(k,2) & \text{if } 1 \leq i{-}M_{1,s} \leq M_{2,s} \\ w(k,1) & \text{if } 1 \leq i{-}M_{1,s}\ {-}M_{2,s} \leq M_{3,s} \\ w(k,0) & \text{if } i > M_{1,s}{+}M_{2,s}{+}M_{3,s} \end{cases}$$

where $M_{4-i,s}$ if the number of nodes of type i among the sons of a node of type s. Recall that the matrix M was defined at the beginning of section 3.

This allows us to write the algorithm in the following form:

> $k :=$ length(n) ;
> **if** $n \leq$ *value*(1^k) **then** $k := k{-}1$; **end if** ;
> $i := 1$; $h := 1$; $s := 1$;
> *border* $:=$ $\boxed{2,\ 12,\ \ldots,\ 1^k2}$;
> **loop**
>   **while** *border*$(k) + \delta(k-1, i, s) < n$
>   **loop for** j **in** $h..k$
>   *border*$(j) :=$ *border*$(j) + \delta(j{-}1, i, s)$;
> $i := i + 1$;
>   **end loop** ;
>   *update*(s) ; $h := h + 1$; $i := 1$;
>   **exit when** $h = k$;
> **end loop** ;

In our example, the successive values of *border* are given by Table 4, below.

Notice that one more step of the outer loop gives us the leftmost node of n. And so, this algorithm gives us both the father and the sons of a node. It is also clear from the algorithm that we know also the status of n.

At this point, we notice that the time and the space complexity of the algorithm are what was announced: cubic in time and quadratic in space. The gain of one power in space is due to the fact that, as the below table suggests it, it is enough to store the last computed value of *border*. The length of border is approximately the length of n and as it stores numbers which are longer and longer and as the greatest one is n, we need a space which is approximately the square of the size of n.

1	8	8	8	8	8
12	68	72	72	72	72
112	672	712	713	713	713
1112	6702	7120	7126	7127	7127
11112	67012	71211	71268	71270	71270
111112	670112	712111	712671	712671	712674

Table 4. Computing the path from the root to a node.

4.2.2 The algorithm for the neighbours

We already know the 'trivial' neighbours of a node: the father and its sons which correspond to white sub-faces of the cell which is assoiated with the node.

We have now to deal with the other neighbours which are hidden by shadowed faces.

Indeed, it is enough to implement the functions *uncle* and *nephew* of our geometrical algorithm. We do this in this section, and this time, the local maps will play the main rôle.

Let us start with an example:

Let us look at the neighbours of the cell (34). An easy computation shows that the sons of (34) are: (272), (273), ..., (278), and its father is (4). The sub-face (34) has four grey sub-faces and it has a single outer brother: the face (29,34). And so, a single one of the grey sub-faces of (34) is adjacent to the outer brother of (34). And so the neighbouring sub-face as a sub-face of (4) is the face (29,34). Notice that the index of (34) with respect to (29,34) is 2. Now we know that the uncle is a brother of the face (4) but namely, the brother which is adjacent to the face (29,34). This cannot be seen neither from the local map which deals with the faces (3) and (4) nor the local map which deals with the face (4) alone. We need indeed a local map which contains both data: in other words, we need a local map which contains two generations, as the map of Figure 11. On this figure, we see that the needed uncle is the face (2). Going further, as we are at the top, we see again that the same local map gives us the nephew: it is the white face of the second generation which is in the face (2) and which is adjacent to the face (29,34). Namely, it is the face (18). We know that this face is a 1-type node which is oriented in the direct order and which is an outer sub-face. Accordingly, thanks to our conventions on the numbering path, we know that the outer face is the sub-face 2, in the local coordinates. And so, we have to look now at the sons of the face (18) which are also easily computed: (142), (143), ..., (148). The local map of the second generation for the face (2), see Figure 13, below, tells us that the black sub-face of the face (18) which is adjacent to the face (29,34) is the face (146,147). Now, the complementary index for the expected white brother is 1 and so, the neighbour of (34) which we are looking for, is the cell (146).

The face (34) contains three other grey faces, and thanks to the local map of the second generation for (4), see Figure 14, these faces are the faces (272,273), (272,275), (275,276), and (276,277). The local map of the face (4) gives us directly the uncles: (33) for (272,273), (31) for (272,275) and (29) for (275,276). The local map of the second generation for (4) gives us the expected neighbours, which are in all cases the outer face of the uncle which is adjacent to the respective grey face of (34): namely, (233) in (29), (251) in (31) and (271) in (33).

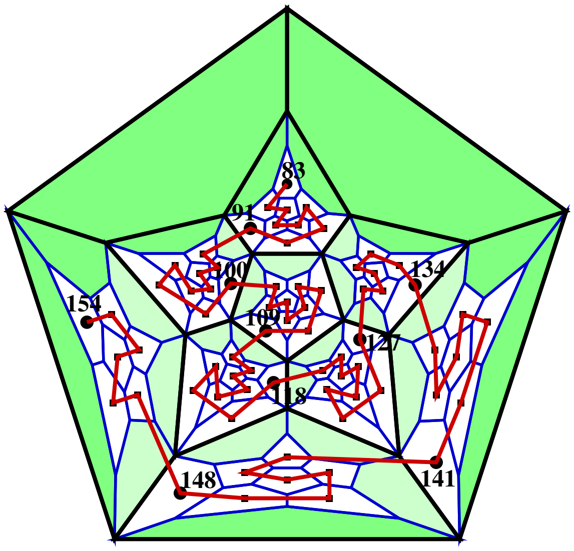


Figure 13. The local map of the second generation for the face (4).

Indeed, thanks to the local maps, it is possible to construct *tables* which allows us to make the correspondance between the uncles, the nephews and the intermediate faces.

Table 5 gives us an example of such a table for the face (4).

On the example, we have seen a single application of the functions *uncle* and *nephew*. As the example of Table 5 is immediately transportable to any face of the same type and of the same orientation by a simple shift, this example is enough to get convinced that starting from a complete set of tables of the second generation for all type of nodes and for each orientation it is easy to compute the *uncle* or the *nephew* of a node.

Thanks to these implementation, we take again the algorithm which we constructed in the sub-section 2.4.1.

face (4)	inner	outer	brothers
(29) inner direct	(227)..(235)	(227,228) (227,232)	(30)<=(239) (31)<=(234) (34)<=(233)
(30) inner reverse	(236)..(244)	(236,241) (236,237)	(29)<=(236,237) (31)<=(244) (32)<=(243) (37)<=(242)
(31) top direct	(245)..(253)	(245,246) (245,250)	(30)<=(245,246) (29)<=(245,250) (34)<=(251) (33)<=(252) (32)<=(253)
(32) inner reverse	(254)..(262)	(254,259) (254,255)	(30)<=(254,259) (31)<=(254,255) (33)<=(262) (36)<=(261) (37)<=(260)
(33) inner reverse	(263)..(271)	(263,268) (263,264)	(32)<=(263,268) (31)<=(263,264) (34)<=(271) (35)<=(270) (36)<=(269)
(34) inner direct	(272)..(278)	(272,273) (272,275)	(33)<=(272,273) (31)<=(272,275) (29)<=(275,276) (35)<=(278)
(35) outer direct	(279)..(285)	(279,280) (279,282)	(33)<=(279,280) (34)<=(279,282) (36)<=(285)

(36) outer direct	(286)..(292)	(286,287) (286,289)	(32)<=(286,287) (33)<=(279,282) (35)<=(289,290) (37)<=(292)
(37) outer direct	(293)..(298)	(293,294) (293,296)	(36)<=(293,294) (32)<=(293,296) (30)<=(296,297)

Table 5. Table of correspondance between faces and brothers for the face (4).

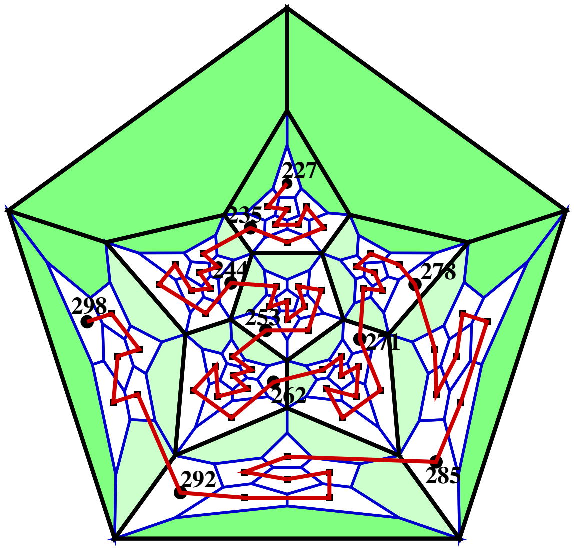


Figure 14. The local map of the second generation for the face (4).

Notice that the transportation of the tables brings a linear term only, both in time and in space, as far as the number of entries of the table is a constant. For the same reason, taking the uncle or the nephew needs a linear time. Consequently, as this is performed a number of times which is at most the size of n, this part of the algorithm for the neighbours is at most quadratic in time. Hence our claim about the complexity of the algorithm.

Acknowledgement

Both author are much indebt to professor Heinz-Otto Peitgen and to the University and the IUT of Metz for making the best conditions for their cooperation.

References

[1] D. V. Alekseevskij, E. B. Vinberg, A. S. Solodovnikov, Geometry of spaces of constant curvature, in Geometry II, E. B. Vinberg ed., Encyclopedia of Mathematical Sciences, vol. 29, Springer-Verlag, Berlin, (1993).

[2] T. Asano, D. Ranjau, T. Roos, E. Welzl, P. Wifmayer, Space-filling curves and their use in the design of geometrical data, Theoretical Computer Science, **181**, (1997), 3–15.

[3] C. Caratheodory, *Theory of Functions of Complex Variable, vol. II, 177-184, Chelsea, NY, 1954.*

[4] H. S. M. Coxeter, *Introduction to Geometry*, John Wiley & Sons, Inc., New-York, 1969.

[5] H. S. M. Coxeter, W. O. J. Moser, *Generators and Relations for Discrete Groups*, II Ed., Springer, Berlin, 1965.

[6] D. B. A. Epstein, J. W. Cannon, D. F. Holt, S. V. F. Levi, M. S. Paterse, W. P. Thurston, *Word Processing in Groups, Jones and Barlett Publ., Boston, 1992* .

[7] A. S. Fraenkel, Systems of numerations, Amer. Math. Monthly, **92** (1985), 105-114.

[8] F. Herrmann, M. Margenstern, K. Morita, Cellular automata in the hyperbolic plane: implementing a polynomial algorithm to solve SAT problem, CA 2000, 6^{th} $IFIP\ WG1.5$ Meeting, August, 21-22, 2000, Osaka, Japan, (2000), 25–26.

[9] B. Grünbaum, G. S. Shephard, *Tilings and Patterns, Freeman , New York, 1987.*

[10] D. Hilbert, S. Cohn-Vosen, *Geometry and the Imagination*, Chelsea Publ. Co, New-York, 1990.

[11] M. Hollander, Greedy numeration systems and regularity, Theory of Comput. Systems, **31** (1998), 111-133.

[12] J. E. Hopcroft, J. D. Ullman, *Introduction to Automata Theory, Languages and Computations, Addison-Wesley Publ. Co, 1979.*

[13] W. Magnus, *Noneuclidean Tessellations and Their Groups, Acad. Press., NY, 1974.*

[14] M. Margenstern, K. Morita, NP problems are tractable in the space of cellular automata in the hyperbolic plane, Theor. Comp. Sci. (to appear).

[15] M. Margenstern, Cellular automata in the hyperbolic plane, GIFM, Nr. 103, I.U.T of Metz, 1999, 33p.

[16] M. Margenstern, Cellular automata in the hyperbolic plane (II), GIFM, Nr. 101, I.U.T of Metz, 2000, 43 p.

[17] M. Margenstern, An introduction to cellular automata in the hyperbolic plane. CA 2000, 6^{th} $IFIP$ $WG1.5$ Meeting, August, 21-22, 2000, Osaka, Japan, (2000), 1–2.

[18] M. Margenstern, A package of algorithms to devise cellular automata in the hyperbolic plane and related questions, Computer Science Journal of Moldova, **8-2**, (2000), 95-115.

[19] M. Margenstern, K. Morita, A polynomial solution for 3-SAT in the space of cellular automata in the hyperbolic plane, J. Universal Computations and Systems, **5-9**, (1999), 563–573.

[20] M. Margenstern, New tools for cellular automata in thye hyperbolic plane, J. Univ. Comp. Sci. **6** 12(2000), 1226–1256.

[21] Margenstern M., A contribution of computer science to the combinatorial approach to hyperbolic geometry, accepted to SCI'2002, Orlando, July, 14-18 (2000).

[22] M. Margenstern, G. Skordev, Locating cells in regular grids of the hyperbolic plane for cellular automata, Universität Bremen, Institut für Dynamische Systeme, report N°455, July 2000.

[23] M. Margenstern, G. Skordev, Rectangular grids in the hyperbolic plane for cellular automata, CA 2000, 6^{th} $IFIP\ WG1.5$ Meeting, August, 21-22, 2000, Osaka, Japan, (2000), 40–41.

[24] H. Meschkowski, *Noneuclidean Geometry*, Acad. Press, NY, 1964.

[25] P. S. Novikov, *On the Algorithmic Unsolvability of the Word Problem in Group Theory*, Trudy Math. Inst. Steklova **44** (1955) (English translation in Amer. Math. Soc. Translations (2), **9** (1958)).

[26] W. Parry, On the β-expansions of real numbers, Acta Math. Acad.Sci. Hungar. **11** (1960), 401-416.

[27] H. Poincaré, Théorie des groupes fuchsiens, Acta Math. **1** (1882), 1-62.

[28] D.M.Y. Sommerville, An introduction to the geometry of N dimensions, Dover Publ. Inc., New-York, 1958.

[29] J. Stillwell, *Classical Topology and Combinatorial Group Theory*, Springer, New York, 1980.

[30] E.B. Vinberg, O.V. Shvartsman, Discrete Groups of Motions of Spaces of Constant Curvature, in Geometry II, Spaces of Constant Curvature, in Encyclopedia of Mathematical Sciences, vol. **29**, Springer, 1993.

[31] S. Wolfram, *Cellular Automata and Complexity*, Addison-Wesley, 1994.