Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Computación

Closure Under Substitution to Proper Flexagons

Tesis que presenta:
Marisol Roldán Palacios

Para obtener el título de:
Licenciatura en Ciencias de la Computación

Asesores:
M.C. Dawe González Arturo Alejandro
Dr. Gómez Soto José Manuel
Dr. Meneses Viveros Amilcar

Puebla, Puebla, Noviembre 2018.

Acknowledgments

I take this opportunity to express gratitude to Ph.D., Harold V. McIntosh for introducing me to this and others unusual topics, for his incessant persuasion to continue our learning and for his unconditional support and attention. Equally, I appreciated his wake-up calls when needed.

For sharing expertise, invaluable guidance and for a constant encouragement extended to me I am particularly indebted to Amilcar Meneses Viveros, Ph.D., J. Manuel Gómez Soto; with a special mention to Ph.D., José R. E. Arrazola R.[a] for his unfailing support.

I am also grateful to each team member of 'Departamento de Aplicación de Microcomputadoras' of the ICUAP (Instituto de Ciencias de la Universidad Autónoma de Puebla) for their help, support and hospitality. A significant mention to J. Pedro Hernández H. and Saúl Zárate V. for their full backing. My gratitude for his willingness to the grammar checking of one of the versions of this work to teacher Arron M. S.

All this effort is entirely dedicated to:

My mentor, H.V.M.[b]
My model, P.G.P.M.[c]
My strenght, J.I.J.R.P.[d]

[a]R.I.P.
[b]R.I.P.
[c]R.I.P.
[d]R.I.P.

Abstract

It is a constant to keep separated physical objects from a possible abstract representation and trying to combine both sides of the same subject early at the beginning of an academic formation is not always achievable due to different factors. A flexagon is an object that anyone has access, to play or to study it, behind them a variety of apparently unrelated topics are combined subtly, a part of them closely related to computer science which motivates this work. Given that the field of study is extensive, a problem was posed, closure property on proper flexagons. Such attribute was approached based on the grammar context and graph theory. In the way of solving the problem in question, the essential elements connected to it were exhibited, among which flexagon maps, graphs, duals, normalization, and language theory were found.

Preface

After Professor Harold V. McIntosh was introduced [McIntosh, 2000a] to flexagons, he had been working on them during different stages of his life. He coined the term frieze code for the abstract sequence introduced and explained in [Conrad and Hartline, 1962] to define a flexagon, just the word frieze is handled as a synonym of the term pattern in the flexagon jargon.

Based on a recursive reasoning process bringing it to its limit, the concept of Universal Flexagon [McIntosh, 2000a] arose, it was called like this figuring out that any other flexagon could be derived from it and compared with the covering space perspective. Identifying the latter which follows a splitting and adding process as a sort of the opposite process of the first. He placed a covering space in the complex variable context as the type of a Riemann surface. Pointing out that if the "splitting and adding" technique is not made suitable to the sections handled, it might result in a tubulation.

From the beginning, McIntosh implemented the use of the maps to look into flexagons, he along with some classmates developed some when exploring flexagons. In actual fact, in the course of experimenting on the joining of flexagons, he examined and tried different types of them to better represent this process. The latter one he worked on, which was presented to me, was the use of a polygon (quite the Jordan Polygon worked by Gross-Lentin [Gross and Lentin., 1967]). It resides in the translation of any flexagon map to a convex polygon which not only facilitates the understanding of the go through step into the triangulated polygon to get the sequence number and consequently the frieze code but, adds clarification to the flexagon's extension by joining.

The basic idea behind it, is to arrange the vertices of the flexagon map on the border of a regular polygon along with their corresponding labels. To the joining, after having identified the side designated for it, we have to extend all of them as long as the sides of the corresponding arrangement of triangles disposed inside our polygons to obtain a consolidated polygon representing the map of the joining result.

As it has happened repeated times in research, McIntosh was not the only one who worked on that regular polygonal representation, Madachy showed a pair of examples which he called "structure diagrams" which worked in a few similar ways [Madachy, 1979]. It is appropriate to point out that McIntosh had the regular polygonal representation before hearing about the structure diagrams of Madachy. Let us observe that, in Madachy's article the generation of a pattern[a] starting at a regular polygon is explained. McIntosh begins with any polygon, convex or not, fulfilling the flexagon map condition then applying a normalization to get the polygonal representation to join them in pairs and, rearranges them to fit half of the polygon to be ready for joining. Termed here as the McIntosh Method after him, due to us being introduced to it from him, the whole process is detailed in section 4.3.

Professor Harold was immersed in different fields of study, inherently, looking for a relationship between each other. By nature, a recursive process brings us to the conception of fractals. He proposed [McIntosh, 2000a], [McIntosh, 2000b] the dragon curves, Hilbert space-filling curve or even, Helga Koch's snowflake to put flexagons in terms of a geometrical fractal platform. And also he handled a Cayley tree as a simplified flexagon map. The time they (he and his students) worked on LISP, although the language theory was being developed in parallel, a parse tree was recognized to be enough to get a flexagon frieze.

---

[a] Frieze

Contents

Introduction

The leading subject of the present work is to show that a flexagon is obtained after merging flexagons in pairs, explicitly manipulating their maps in graph terms. That means, to exhibit the closure property to flexagons under joining operation. The grammar context was identified as part of the tools to achieve this. Analyzing the use of polygons along with the closure property for context-free grammars gave us the bases to figure out the solution to have the main achievement. The use of maps has been encouraged, in this case, a Jordan polygon plays the role of a flexagon map.

The structures called flexagons can be studied from different aspects. As a pure entertainment perspective as well as part of a formal mathematic field, or even as an artistic regard. The time[-1940] in which the flexagon committee (Stone, Tukey, Tuckerman Jr., Feynman) was analyzing those structures at Princeton University some kind of them had already been patented. Even though there is certain evidence that flexagon committee worked on a paper summing up the properties they found out about the behavior of flexagons they analyzed, that document was lost in history under the hard circumstances of their time[1].

Beside the facts stated above along with other events were combined such that nowadays the interest in them is not only alive but growing. Among contributors we can cite: Stalker[1935], Rutherford[1939], Tuckerman Sr.[2], Rogers and D'Andrea[April 1955], Garner[December 1956], Oakley and Wisner[March 1957], Wheeler[1958], McIntosh[[McIntosh, 1960] referred in the work [Conrad and Hartline, 1962]] , Conrad[1960], Hopkins[July 1961], Conrad and Hartline[May 1962], Pook[2003,2009]. In August 2005, the flexagon Lovers yahoo group [Schwartz, 2005] was founded by Ann Schwartz. Years back, the group worked very dynamicaly, at the present it looks quiet, however, their members continue actively involved in personal or group projects, part of them maintain their own web pages [Moseley, 2007], [Pook, 2012], [Schwartz and Schwartz, 2002], [Sherman, 2000] full or partially dedicated to flexagon topics.

Flexing and rotation were the first characteristics related to flexagons. Triangles as base polygon[3] are among the most studied, nonetheless, soon the same properties concerning triangles were analyzed to make certain that they still verify for n-gons. In fact, a concept of an ultimate flexagon was introduced by Conrad 'n Hartline in their work [Conrad and Hartline, 1962]. The idea is that it represents a kind of universal flexagon, from which any other can be derived.

The ultimate flexagon was defined in math terms as a covering space, so strictly the flexagons obtained from it could be seen as subspaces of it. They proposed a circle representing an n-gon, the covering space will do as follow: first, apply a three arcs' division, then each arc will be substituted by two others. Depending on what flexagon you like to get, the latter step is applied conveniently as many times as it is needed.

The ultimate flexagon is a deductive treatment, however, to achieve that conclusion an inductive procedure was first developed. The present work is partly dealing with the inductive idea. The main task assigned, was to show that the result of joining flexagons maps in pairs is attained a new flexagon map. For such purpose, flexagons were put under graph theory terms, not before

[1]The beginning of the WW II.
[2]Time he worked at the National Bureau of Standards -US.
[3]Also referred as elementary polygon.

the flexagon concepts were clarified and illustrated whenever possible. In addition, a short set of properties considered closely related to flexagon's extension was included.

To understand the operation of attaching, the frieze code conceptualization and the procedure of how to get it was widely explained. At this point, it is appropriate to clarify that joining, attaching, merging up and, linking terms are indistinct used when joining operation is referred to. A frieze code, the two array of numbers along with a sign sequence, defines a flexagon. Once the concepts derived from the handling of flexagons were put in the terms of graphs, the joining is presented based on the use of the polygon system, a normalized form of its correlated graph.

In addition, to appreciate the computational approach even more, the necessary theory to correlate flexagons with language theory is covered. Relevant due to the substitution closure property showed in terms of a defined grammar will support the conclusion that the result of the joining of flexagon maps will produce again a flexagon map. But not before clarifying the correlation substitution operation in grammar context with the joining operation of flexagon maps in pairs.

Being familiar with what a frieze code is, is basic in order to go through with our objective, to extend flexagons linking them. Due to this its definition was removed from the definitions section dedicating a segment to describe it in full detail. To clarify how the frieze code joining works, among the initial designations was to do a program for achieving their joining. Later, once the linking process was understood, the frieze code concept is introduced. The essential of the graph and grammar context were presented to complement the background.

Showing how recreative structures as flexagons are closely related with fundamentals we face in the computation theory, is a good approach to introduce students to more complex concepts, specialized relationships or even more, to go for a profound study once we have the bases. Nowadays, graph theory is at a mature level, it allows us to try to identify existing relationships between flexagons and fields of study such as fractals (trees), languages theory (graphs) among others.

This work was organized into six chapters, Chapter I is dedicated to set the context of flexagons, what works were previous in general and particularly the ones focused on the use of an abstract representation of them, basic concepts and properties related to flexagon extension. Chapter II gives us the necessary elements of graph context to associate it with flexagons. Chapter III sums up definitions of language theory and its relation to triangulating a surface is introduced and explained which gives the bases to have Catalan Numbers. In chapter IV, the frieze code concept is explained in detail, in addition to the complementary notes of flexagons' abstract representation. The McIntosh method for solving the joining of flexagon maps in pairs is presented. Chapter V is dedicated to show the closure property to flexagons, reinforcing the connection at the end between Universal Flexagon to other elements. Four appendices complement detail. Appendix A is a supplement of properties. Appendix B is showing pictures of McIntosh's studies on maps representing flexagons as well as those introduced in section 4.3. Appendices C and D explicitly illustrate the set of parse threes explained and one more example complements explanation in chapter 5.

Chapter 1

Flexagon context

In this chapter, the preliminary elements to be involved in flexagons are presented. The first section informs us about a part of the history bearing in mind that we do not have all the elements to establish the beginning of flexagons with certainty. The purpose of including the original pictures is to give the credits to whom they correspond, showing how advanced the previous works were, although the terminology to name them was introduced later.

The purpose section was included here along with some prior work which uses maps in joining, basic in understanding them as a tool of our procedure to achieving our aim. The equivalent map types were added to the original diagrams of Wheleer[Wheeler., 1958] as presented by Conrad and Hartline[Conrad and Hartline, 1962]. Flipagons were analyzed to confirm if they were related to our work, they can still be thought of as an extension of flexagons, however, are far related to what we are doing in the sense that the result is a kind of extension but flexagons are not actually linked but chained. Furthermore, a bunch of properties are contained to set some bases on which we will work later on, broadly they are due to the substantial work of Conrad 'n Hartline[Conrad and Hartline, 1962]. We focus on those attributes related to the effort of extending flexagons, nonetheless, not only Pook but some other members of the flexagon lovers yahoo group have been working in establishing general properties to flexagons.

1.1 Intro & Purpose

1.1.1 History

Talking of flexagons, it is not easy to provide an accurate date in which the first one was modeled, even with all the information available nowadays, however, from time to time, individual or group efforts have been made based on diverse sources of inspiration.

For instance, there is a variety of records in the U.S. and International patent's databases related to flexagons where they have been seen as an item of publicity but mainly as a sort of funny toy. We can even find records in the registered trademark database of other countries like France and the U.K. But before we continue talking about flexagons, for those who do not know about them, we can say briefly[1], that they are structures involving folding in their assembly and that we basically operate them with flexings and rotations.

Going back in time, one can find "advertising medium or toy" [Stalker., 1935] a work of Ralph M. Stalker which was filed on April, 1933. There, we can appreciate what in recent years have been treated as flexagons based on a three dimensional tetrahedron body as figures A, B, and C illustrate in 1.1, original pictures were borrowed from [Stalker., 1935]. Stalker is describing a structure made of a single[2] sheet, whose hinges are arranged in a specific way giving it flexibility.

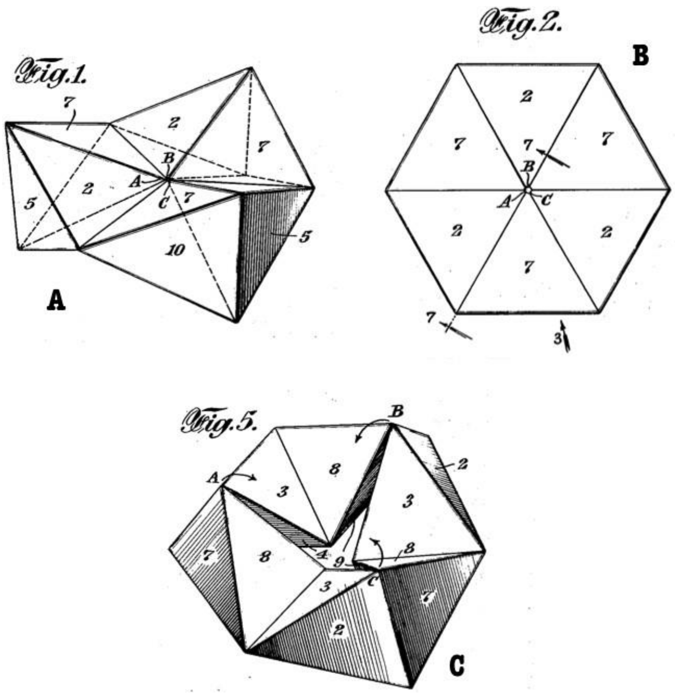

Fig. 1.1: Flexagon. - Stalker-1-2-5

Less than a decade later, we have the work "Toy" [Rutherford., 1941] of Norman. F. Rutherford which is documented in patent US2,245,875, filed on December, 1939. According to this work, "Toy" was thought of as an improvement on what it is known as Jacob's ladder. The plan[3] illustrated in figure[4] 1.2 is using a pentagon as its base polygon, the toy completely assembled is

[1]Meantime you have one in your hands.
[2]Do origami and flexagons, in a certain sense, belong to the same family?
[3]Also named pattern, it will be defined later.
[4]Original picture was borrowed from [Rutherford., 1941] -

shown in fig. 1.3. Now using a square as the primary figure, we present the next group of original pictures 1.4 which were still borrowed from Norman Rutherford's paper [Rutherford., 1941]. The use of only one label (in fig. 1.4, B) in place of two to differentiate the upper from the lower side could give the impression of mixed faces, which is not the case.

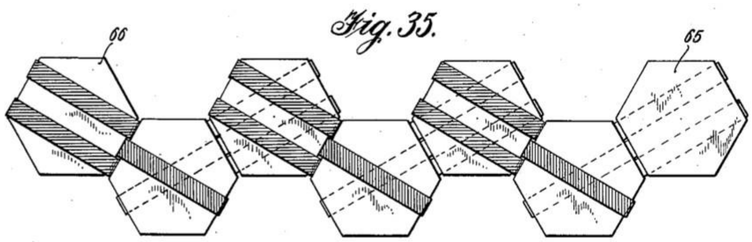

Fig. 1.2: Penta-flexagon. - First known plan - Rutherford 35

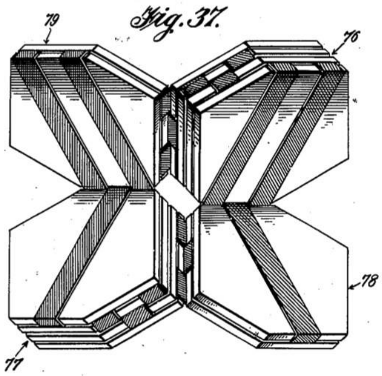

Fig. 1.3: Penta-flexagon. - First known assembled - Rutherford 37

From beginning to end, Rutherford's writing recurrently uses the term face which later became a basic concept in flexagon's context. More so, his phrase "the cross hatching throughout . . ." exactly describes what is taking place during flexing operation. Inspired by Jacob's ladder, this couple of samples are currently known as Penta-flexagon and Tetra-flexagon, respectively.

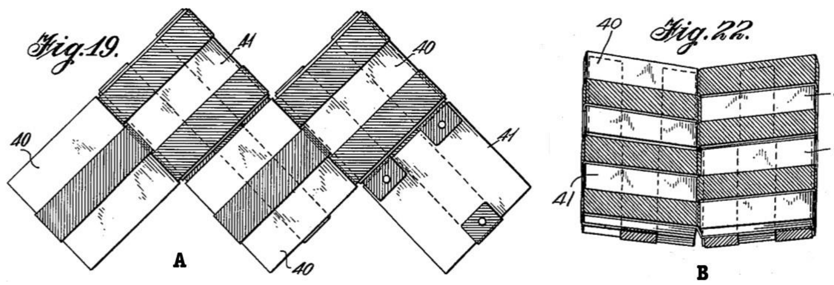

Fig. 1.4: Tetra-flexagon. - plan 'n position 3 - Rutherford 19-22

Once more, it is pointed out that this latter work was filed in 1939 to remark on the fact that it was going parallel to the events happening at Princeton University. First, it was Stone, then the flexagon's committee after which fever for Hexa-flexagons was widely spread for all the university's corners. The committee referred to was formed by Feynman, Tukey, Tuckerman (Jr.) and Stone. Focusing on the analysis and understanding of flexagons, their dedication was interrupted by the World War II. Little by little each investigation or people with enthusiasm have contributed in one way or another to the study and understanding of flexagons.

A little later, the document with patent number US2,245,875 [Rutherford., 1941] along with others contributed to the work of R.E. Rogers et AL [Rogers and D'Andrea., 1959], it was filed from February, 1955. There we can find different friezes to build Hexa-flexagons. We worked on some of them to show here. Figures 1.5.A, B and, C illustrate preparation, figure 1.5.D shows the object assembled. Finally, figure 1.5.E exemplify the flexagon after some flexes.

Hopkins [Charles.L.Hopkins., 1961], more than providing a definition of a hinge, explained the functionality we can get with it. For instance, to enable sides moving back and forth, as well as to rotate the whole assembled structure around a series of axis. "Twisting all sides of the modeled object" about its hinge axes is obtained all at once. Furthermore, he analyzed how many different faces are obtained as a result of that operation. All above was done before August, 1956 when the paper was filed.

1.1.2 Spreading

Now we will figure out how flexagon knowledge was spread even though at the beginning efforts were isolated. The first contributor was Louis B. Tuckerman (father of Brian Tuckerman[5]). During his time in the Westinghouse Science Talent Search[6] event, for a while he introduced to flexagons to the winners of that event. Then, in January of 1957, after Gardner interviewed [Barcellos and Gardner., 1979] John Tukey and Bryant Tuckerman, the article about Hexa-flexagons appeared published in the Scientific American magazine.

In March, 1957 The American Mathematical Monthly published Flexagons which was the first theoretical analysis report [Oakley and Wisner., 1957]. Oakley and Wisner went beyond to a descriptive level where Flexagons were mathematically analyzed. Three years later, the Theory of Flexagons work by Anthony S. Conrad was put forward [Conrad., 1960]. Later, in 1962 the work Flexagons by Anthony S. Conrad and Daniel K. Hartline was concreted [Conrad and Hartline, 1962]. The latter two were printed as RIAS[7] reports.

As a toy or as a subject of more dedicated studies, the interest on flexagons has been growing. We can find different sorts of divulgation. Different personal webpages are being placed online sharing discoveries or rediscoveries about flexagons, among them we can find [Moseley, 2007], [Pook, 2012], [Schwartz and Schwartz, 2002], [Sherman, 2000]. Some of the efforts shared or published have gone into more depth. The flexagon lovers, the group hosted on yahoo, was created some years back. Gathering4Gardner organization, created [(G4G), 2012] years ago after Gardner. Each period of time they organize an event where we can find a variety of subjects addressed to encouraging analytical thinking bearing in mind the funny side of science. A. Schwartz, S. Sherman, T.B. McLean, C. Sonenshein, E. Iacob, J. Beier, C. Yackel, T. Anderson, H. Pajoohesh, C. Smith had joined to the speakers of G4G which have been presenting their work's results, these range from classic hexaflexagon and their regular faces to a new sort of flexing or even to the algebra of tetraflexagons and more.

1.1.3 Prior work joining using maps

Conrad had already given a first approach of linking flexagons in his work "The Theory of the Flexagon" [Conrad., 1960]. There, "the traverse flexagon graph", although not termed in that

[5]Flexagon's Committee member.
[6]U.S.
[7]Internal RIAS report.

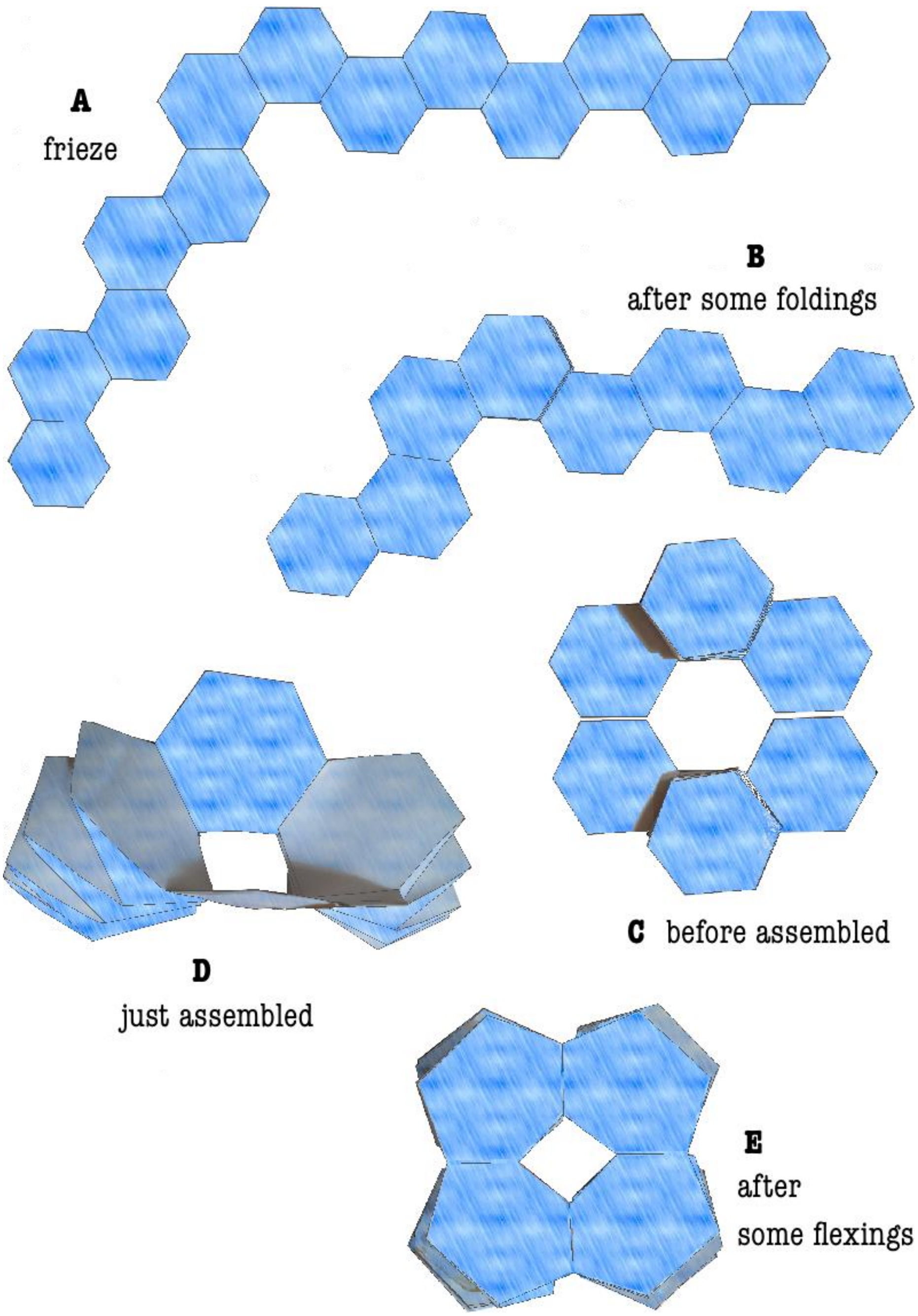

Fig. 1.5: Hexa-flexagon - frieze - folded - assembled - flexed - Roger et Al fig4G

way but can be related to, is used to explain and clarify the resulting behavior. For each triangle represented on the map, one side on the flexagon is intended.

In terms of flexagon maps, one main result on this subject is that it is possible to add in new sides only on the map's border. And then, although side by side and one after another, for the first time it is declared that it is possible to add[8], to one existing flexagon, as many sides as one wants. The technique to extend flexagons called splitting is explained in more detail in Conrad and Hartline's work [Conrad and Hartline, 1962]. Notice that the splitting is handled on the pattern, in the map we can naturally see it as a joining.

In the Mathematical Gazette [Wheeler., 1958]; Wheeler worked on maps that can be appreciated quite similarly to that appearing in the Conrad and Hartline paper, easily we can get the equivalence. The right side of picture 1.6 illustrates the equivalence for the first map presented in Wheeler's work (left side). In his article, Wheeler explicitly showed the splitting process for the Star[9] flexagon map case. Figures 1.7 and 1.8 illustrate a couple of more examples of equivalences between Wheeler's and, Conrad and Hartline's maps. Smalls mistakes in the tags had been identified when generating the map referred to in M4" [Wheeler., 1958] which were pointed out and corrected in figure 1.8.

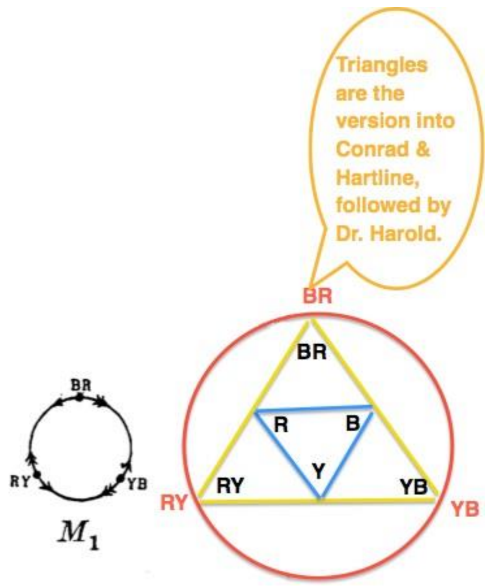

Fig. 1.6: Hexaflexagon - CircleDiagram - Wheeler M1

L. Pook has been working on linked flexagons, the procedure he used is much the same as what was already explained by Conrad and Hartline, Pook names it as splitting in the flat position. It is when the pile has the form $(1, n - 1)$ or $(n - 1, 1)$, i.e. when it has single pats in flat mode. Furthermore, Pook works with box, skew and slant positions. In compliance with the categorization he made.

Based on a relationship defined between the edge flexagons, flexagons with edges as hinges and point flexagons, flexagons whose hinges are now hypothetical points, along with the fact that it is possible to extend edge flexagons linking them, so he declared that theoretically linking

[8]Linking.
[9]Not mix with star flexagon in terms of Sherman or Pook.

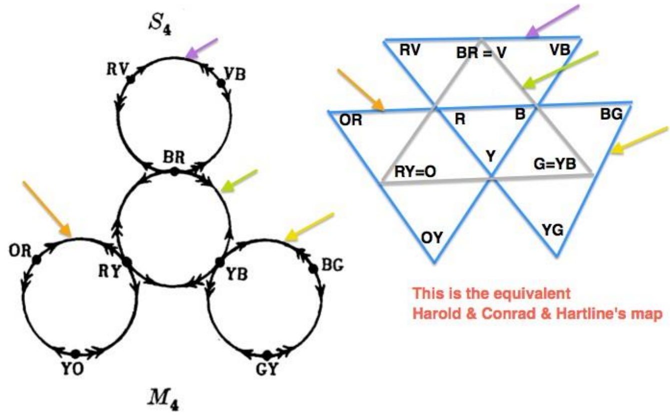

Fig. 1.7: Corresponding C'nH - CircleDiagram - Wheeler M4

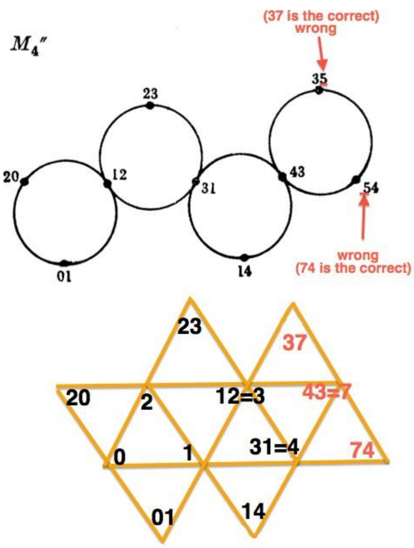

Fig. 1.8: Corresponding C'nH - CircleDiagram - Wheeler M4"

also operates in point-flexagons and that this can be extended indefinitely[10].

After applying a linking[11] operation, when we are collecting labels to generate the frieze code we could realize that the piles corresponding to the nested parts have suffered reversion, Conrad and Hartline identified and explained this phenomenon in their work, more recently Pook also did. McIntosh outlined the recursive linking process in his article A Quick Flexagon Survey [McIntosh, 2000b]. Furthermore, he also observed the convenience of making a map to analyze the replacements as well as to interpret the resulting relationship. The latter work handled complete flexagons as replaceable elements.

The use of the lowest level map was suggested, the Tukey Triangle Network [fig. 1.19] which not only has orientation information but:

1. It takes into account what is visible at the flexagon, both:
   i) On top of.
   ii) Underneath.

2. It reflects both:
   i) The base of a cycle representing a polygon stack.
   ii) The recursive step of removing an elementary polygon and replacing it with the counterspiral of - the incoming - flexagon to be joined[12].

When substitution (splitting then joining), it is observed that orientation has to be held on the side it happens. More recently the Tuckerman Traverse diagrams shared by Moseley [Moseley, 2007] could be appreciated as the result of joining some two others. However, due to the maps or the patterns from which such diagrams were obtained are not shown, i.e., those of the original flexagons, the fact that such Tuckerman Traverse diagrams were not conceived as a linking or as an extension of some others is also possible.

Ralph Jones, has been analyzing the splitting process, attending part of the work of Les Pook, and he had presented some final diagrams [Jones, 2013] of flexing trying to give the flexagon's definition in the terms that Sherman handles. Scott Sherman has been working on splitting [Sherman, 2000] although, for a theoretical handle, he is recommending the Conrad and Hartline paper [Conrad and Hartline, 1962].

Finally, we checked slipagons, a set of them is examplified in figure 1.9, they are basically chains of hexaflexagons. After we assembled and analyzed some of them we conclude that they can be appreciated as a kind of extension of flexagons, however, they had no relation to the joining concept handled here.

1.2 Flexagons's context concepts

Definitions were grouped in this way figuring that they can be referred to them easier. Those who still know the terminology can skip it or can find at a glance whatever they need to consult. We tried to include each existing concept considering the previous as well as recent terms doing the corresponding clarification. Every time it has been possible, we include representative existing pictures some with subtle modifications to make clear what they are illustrating or some new ones were elaborated trying better explain the concepts.

A **flexagon** can be defined in multiple ways. Let us think about it in terms of **a group of piles of hinged polygons**. Hexaflexagon is a well known example; in such case we have six piles of triangles; each one of these is called a pat. Each element (polygon) of the pat is called a leaf.

[10] As Conrad and Hartline priorly predicted to edge flexagons.
[11] Joining.
[12] Linked.

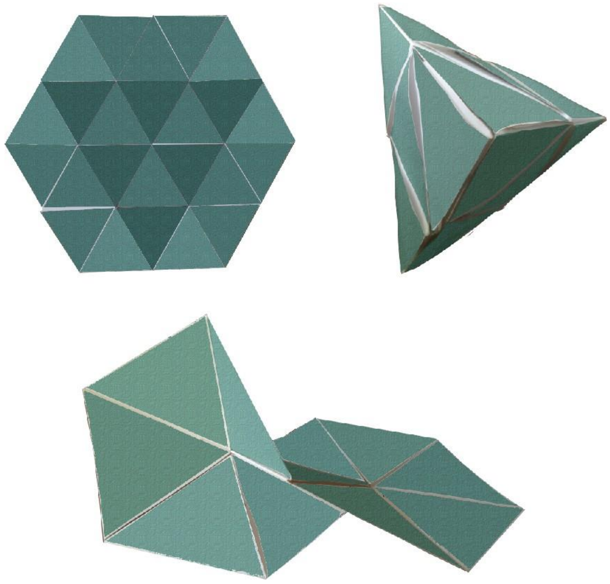

Fig. 1.9: Example - slipagons

Unit = **one leaf thick pat + (adjacent)** n - 1 **leaf thick pat**. The space between two pats and a hinge where it is possible to pass ones' thumb is named a thumbhole. Convenient titles were added in figure 1.10 to exemplify concepts above. A pat can be whichever, a single thickness or, from a recursive perspective, the merge of minor reversed pats.

According to the flexagon in chance, we have to complete the appropriate steps to go from one face to another. Imagining the action of *" the cross hatching throughout . . ."*[Rutherford., 1941] could give us a better idea of what is happening turning a flexagon in such a way that one can expose a hidden face. That kind of manipulation is called flexing. Early, **pinch flexing**[13] was defined, this is illustrated in figure 1.11. Nowadays, a variety of flexings have been considered to operate on flexagons, Sherman [Sherman, 2000] explains them all in detail.

The term plan refers to **the pattern of the paper strip** from which the flexagon is folded up. The appellations layout, pattern and net allude to the same. More recently the term frieze was incorporated to address the same thing. The plan is obtained based on the conjunction of the sign and number sequences. The class is the number of sides of the polygons in the **plan**.

The regular flexagons set comprises those which come from **"straight strips"**, the rest of them will be classified as irregular. Even though, the differences are merely certain easiness and symmetry in operation that regular flexagon posses.

The group of leaves physically arrayed together up or down could make reference to the face, and also to the side, of a flexagon. However there is a subtle distinction between them. The leaves assembled together are exactly the same but not in the same *"accommodation"*, the simplest example is the smiled/sad flexagon; two faces were got from the same side. Note that such variation comes from the angle of rotation.

For handiness, each vertex of a triangle is associated with each side listed below:

1) **the side on the back of the flexagon**

[13]This document is restricted to pinch flexing.

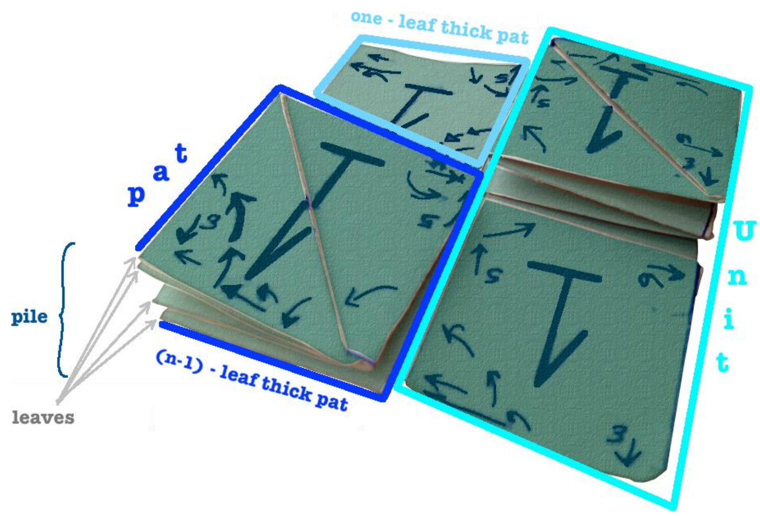

Fig. 1.10: Ex-basicConcepts-tetraF

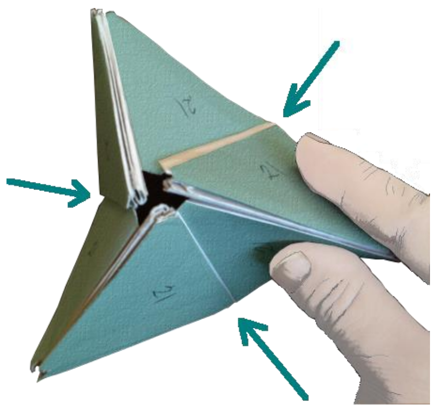

Fig. 1.11: Ex-pinchFlexing

2) the side on the front
3) and the side which comes up next

in the way that figure 1.12 exposes. The rule of thumb [Conrad and Hartline, 1962] names this relation.

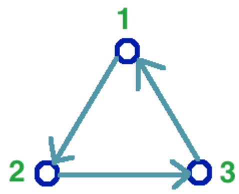

Fig. 1.12: Map: the back - the next - the front

Setting this relation results in a map[14] of the flexagon. Which uniquely determines the flexagon by means of **an array of triangles**[15]. Cycle is the number of sides of the polygons on the **map**. Provided a map, the branched structure obtained replacing each triangle[16] by its corresponding midpoint and then connecting them; is called Tuckerman Tree [17] (Fig. 1.13).

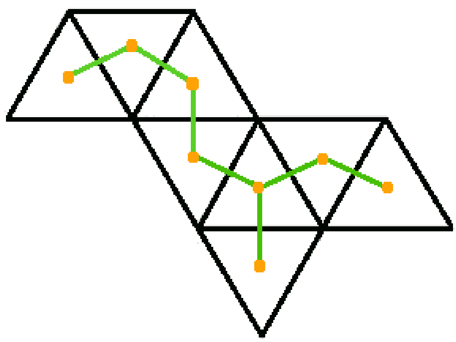

Fig. 1.13: Dual - Tuckerman Tree

In this context, the term path refers to a line segment joining two adjacent points in the tree or on the map. Once the path concept has been introduced. Formally flexing operation is said to be represented by traversing the paths on the map, it is the change from one path to another. One should realize that each **path** represents the position of the flexagon in which the sides at either end of the path in question corresponds to the sides exhibited on the front and back of the flexagon in the given position (or in a given path). Figures 1.14 and 1.15 exemplify this.

No matter what sort of faces, flexing is established as replacing one side by a novel one, the time the other is moved. An operation push through is that exchanging the dispositions of the flexagon's top and down sides in such a way that each arrangement folds together. A roll is an uninterrupted sequence of flexes, along a given hinge. **The number of leaves in the plan of one unit** equals the order of the flexagon which matches with **the number of vertices in the map**.

[14]Some times referred as the -go through- flexagon map
[15]In the sense of a set of triangles disposed in a determined way.
[16]The general case is an: n-gon.
[17]After Tuckerman Senior, its inventor.

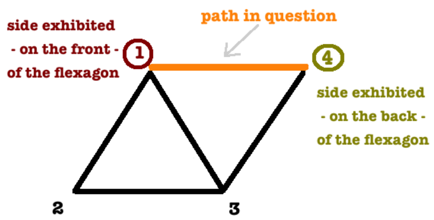

Fig. 1.14: Ex-1 correspondence: path - flexing

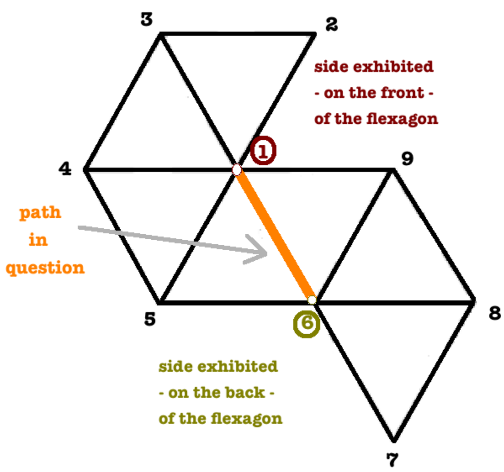

Fig. 1.15: Ex-2. - a path in the map go through is - flexing

Plan's system of orientation, is the set of **head to tail** arrows drawn about the sides of one flexagon plan (not strictly in triangles)[Conrad and Hartline, 1962]. This system of orientation along with the original "+" -positive- sense, arbitrarily set, will determine if a triangle (square, pentagon, ... n-gon) is called "+". When traveling along the plan, if the polygon is left in the same direction as that indicated by the arrow **crossed over when entering into it**, or "-" if it is left in the opposite sense. This is used in the generation of the frieze. Figure 1.16 exposes that. The diagrams 1.17 A, B, and C add clarity to this notion.

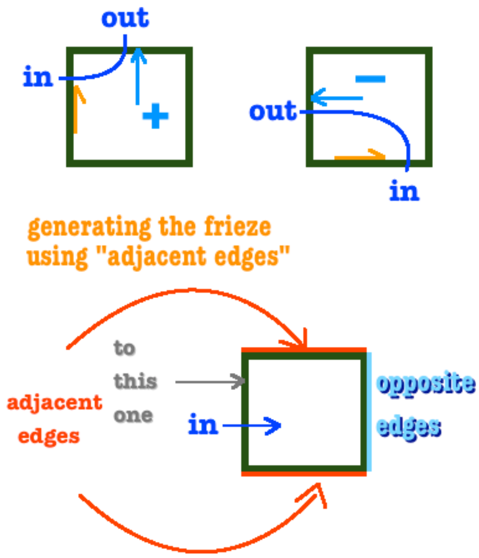

Fig. 1.16: Joining rule - adjacent sides - positivoNegativo

The Tukey triangle network is equivalent to the hinge network. One line in that net summarizes the use of one hinge for a number of flexes, which is the number of paths (in the map) traversed by this line (in the network). The course of the path described by **the Tukey triangle network** strictly traces the **Tuckerman traverse**. Pictures 1.18, 1.19 instruct **how to get the Tukey Triangle Network from a map and exhibit how it looks**. Draw 1.20 [Conrad and Hartline, 1962] illustrates how to pro- duce the plan of a flexagon using oriented triangle system in conformity to a presented sign sequence.

The Oriented Triangle System imitates a board for playing checkers, shadowing alternative triangles, given an oriented triangulated network, (fig. 1.21). That leaves arrows about shadowed and light triangles advancing in opposite direction to each other. And shadowed triangles placed on a vertex in contrast to light triangles which rest on a base. The achieved oriented systems to each dark and light triangles support a new method to ascertain the course (positive/negative) of a triangle on a grid. A tool **to generate the plan**.

Every external vertex in the Tukey Triangle Network takes place at the midpoint of a map line. This line can turn out in one of three directions which are spaced 120^o apart. Adding the addressing to the map (fig. 1.22) indicating the orientation of the line going through the **Tukey Triangle Network**, i.e., the **Tuckerman Traverse**. Following the Tukey Triangle Network according to the map direction while collecting the signs in each vertex is the procedure to have the sign sequence. A symbolic depiction of the flexagon fundamental to accomplish the flexagon plan. Triangles are the classic approach, nevertheless the concept could apply to a map formed of any elementary polygon[18].

[18]Bearing in mind the graph perspective.

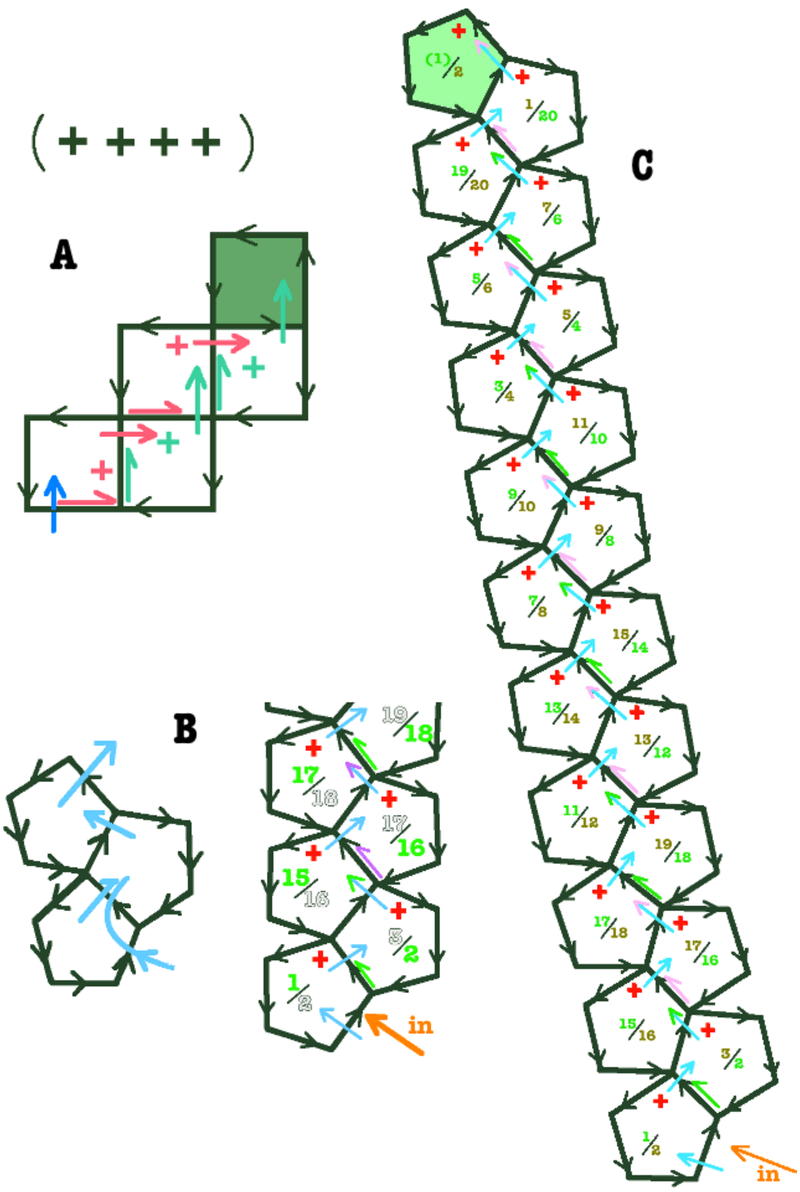

Fig. 1.17: Example - square - partialPenta - penta2

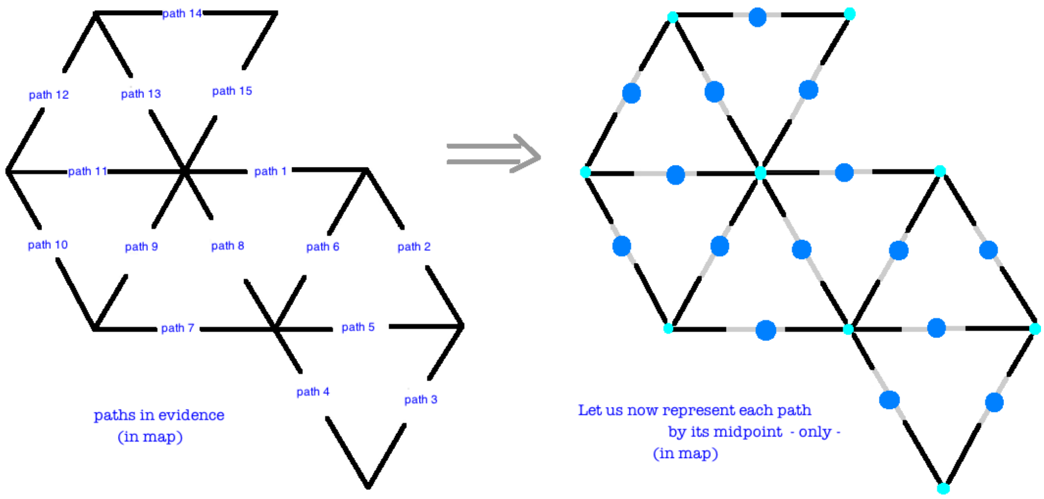

Fig. 1.18: TTN - inEvidence-paths-midpoints

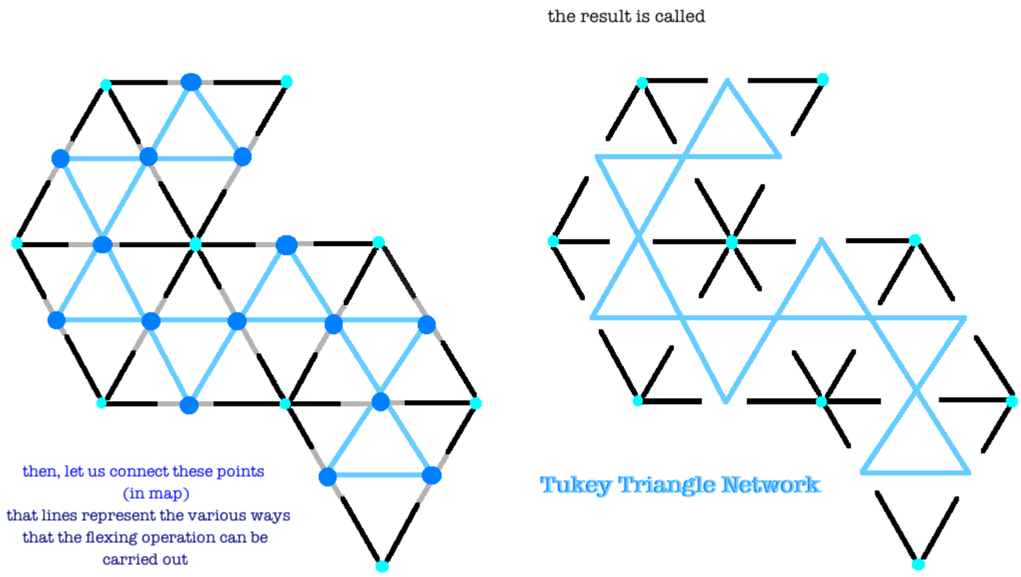

Fig. 1.19: TTN - connected lines 'n the result: the Tukey Triangle Network

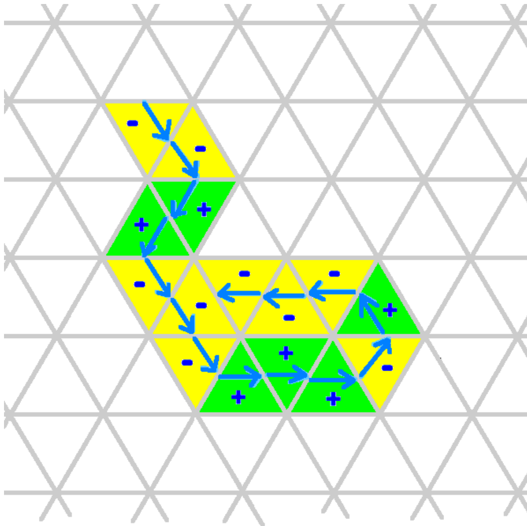

Fig. 1.20: Triangle System - plan

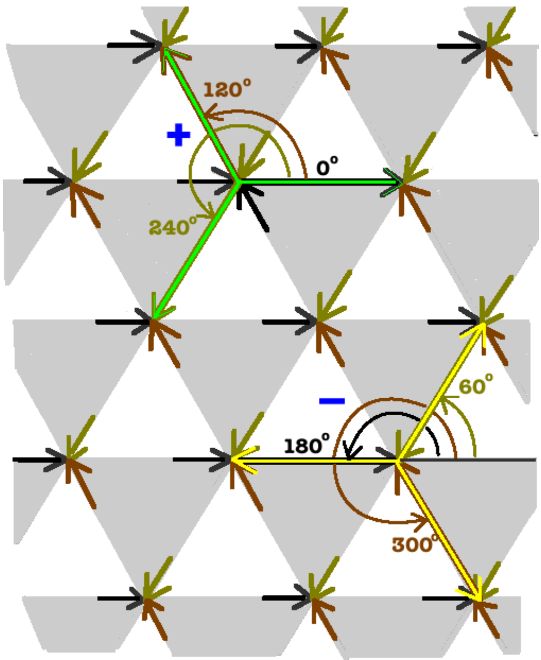

Fig. 1.21: Oriented Triangle System

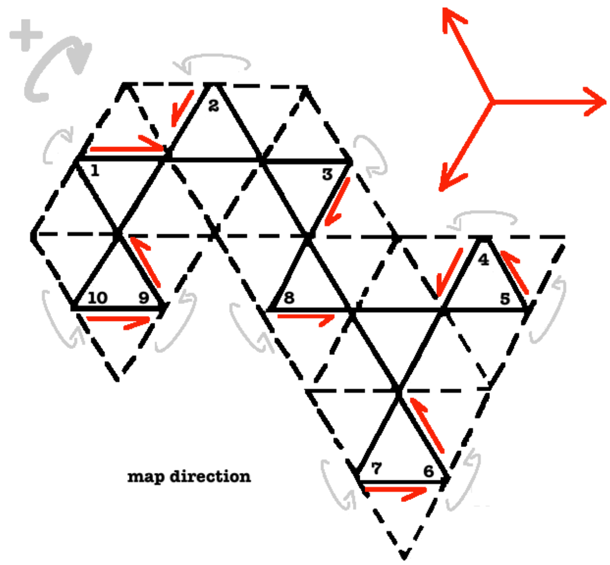

Fig. 1.22: Map addressed

Among several forms of an abstract representation of a flexagon, we find: The map. Tuckerman tree. Tukey Triangle Network. Tuckerman Traverse. Sign sequence. Frieze code. Number sequence. Constant ordering number and the plan.

Because the Frieze Code is a fundamental concept in the present work, we postpone a detailed introduction and the method to generate it in section 4.1. The constant order and the basic number sequence are inverse or dual to each other. We have the number sequence collecting the tags according to the path traced in the polygonal representation once its vertices were consecutively labeled clockwise or counterclockwise. The constant order will consist in numbering the polygon's vertices consecutively but following the internal path.

It is convenient to observe that a fairly identical principle to generate the Tukey Triangle Network could be applied to manufacture the Tukey n-gon Network. Illustration 1.23 is a sample. Furthermore, given one of the three elements, the map, the tree, or the Tukey triangle network; the other two are attainable.

In this point, it is possible to extend N, the order behind the flexagon's definition. It equals to:

- The number of vertices on the map.
- The number of lines on the tree plus 3.
- The number of exterior vertices on the triangle network.
- The number of signs or pairs of numbers(up/down side) in the two sequences.
- The number of leaves in the unit plan.

Given a number sequence, the next steps generate the Tukey Triangle System:

- identify the minimum *min* and maximum *max* of it,
- numerate from *min* to *max* consecutively around an imaginary circle, do it equidistant,
- then connect them in accordance to a given sign sequence.

Tukey Triangle System is pictured in 1.24. When it is generalized, the Polygon System is obtained.

Reduction of the sign sequence method. The method is governed by the following rules. Rule 1, only adjacent signs in the sign sequence are reducible if they are still adjacent in the map,

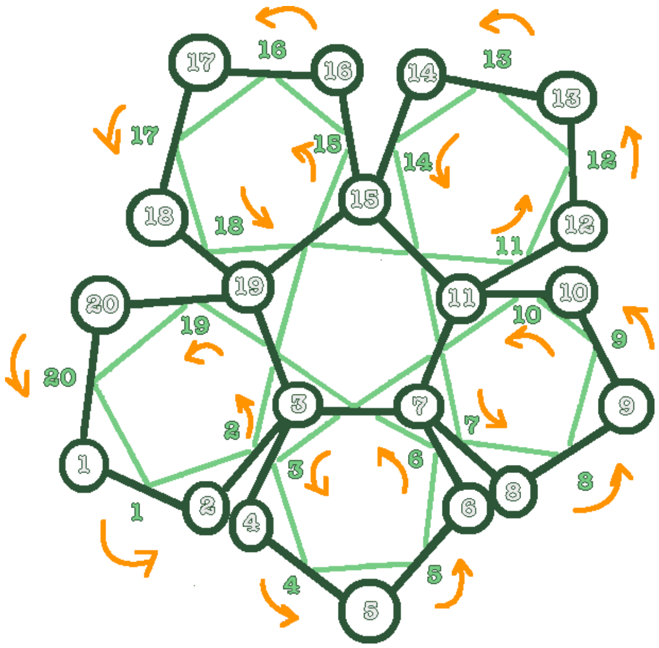

Fig. 1.23: Ex-TukeyPentaNetwork

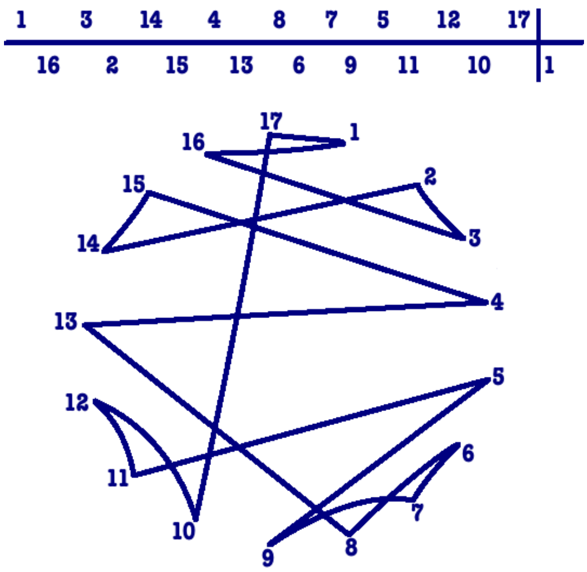

Fig. 1.24: Tukey Triangle System coco - Tukey Triangle network correspondence

otherwise it doesn't make sense. Rule 2, if the reduced sign sequence is that of a declared flexagon; then the original one also characterizes a flexagon. Which is useful to identify if a long sign sequence is a valid flexagon.

A Flexagon family, a bunch of flexagons sharing some attributes. The shape of their maps resolves next classification:

Chain flexagons family. Generated by coupling end to end the map unit (fig. 1.25.A). Figures 1.25.B and C contain its characteristic map and its Tukey Triangle network respectively. Its characteristic sign sequence + + - + - + - + ± ∓ ± ±, the last couple of signs depends on the order, if it is even or odd.

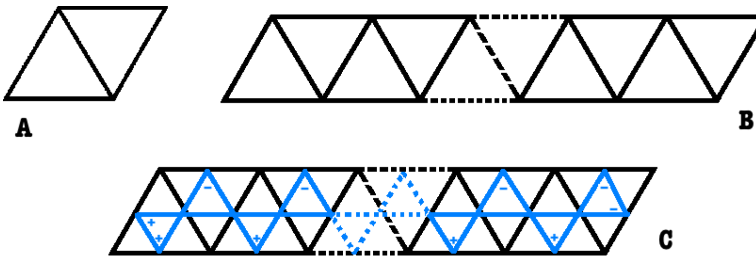

Fig. 1.25: ChainFlexagonFamily UnitMap CharacteristicShapeMap TukeyTriangleNetwork

Fan flexagon family. Based on the same map unit but joined around a vertex. Characteristic form depicted in picture 1.26. Characteristic sign sequence: + + + + + ... ± ± - - - . . . Middle signs depend, once more, on the order of the flexagon.

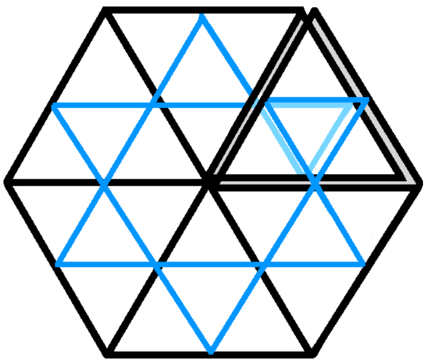

Fig. 1.26: FanFlexagonFamily-CharacteristicMapShape

The star flexagon family. The natural example is produced by successive uniform doubling of a straight strip, figure 1.27 clarifies how. Double successively as many times as levels will be worked. The characteristic map in (fig. 1.28.B). Drawing 1.28 approaches it better.

Based on the above classification, it is anticipated that each flexagon map can be typified as a composition of the chain, the fan and the star kind.

From the star family it is possible to define the universal flexagon. Basically, the universal flexagon is the flexagon containing all **possible** sides that a flexagon could contain. A specific flexagon could be derived from it by deleting identified[19] sides. In order to understand the

[19]Necessary.

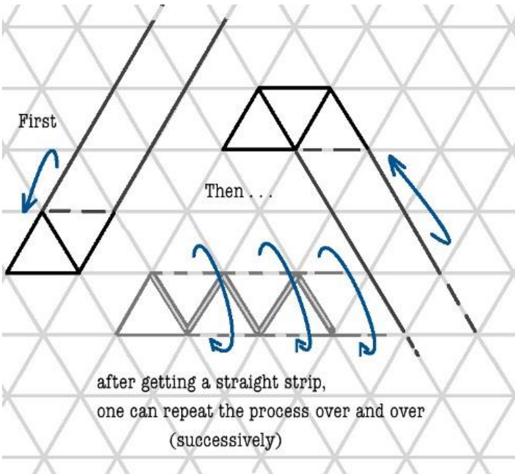

Fig. 1.27: StarFlexagonFamily-howToDouble

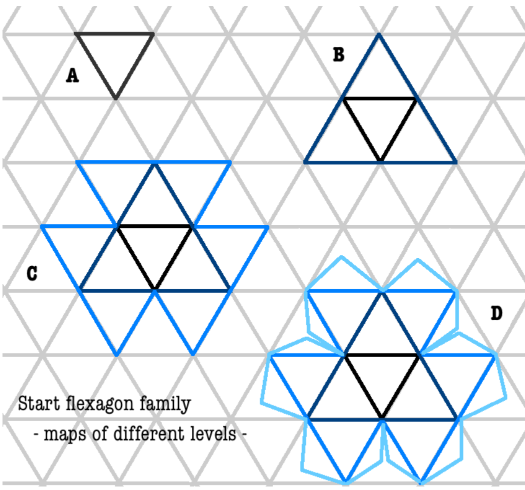

Fig. 1.28: StarFlexagonFamily-CharacteristicMapShape-difLevels

concept better, illustration 1.29 supplements the Tukey triangle networks, corresponding Tuckerman Trees are pictured in 1.30, 1.31. Note that this is the idea behind the arched covering space.

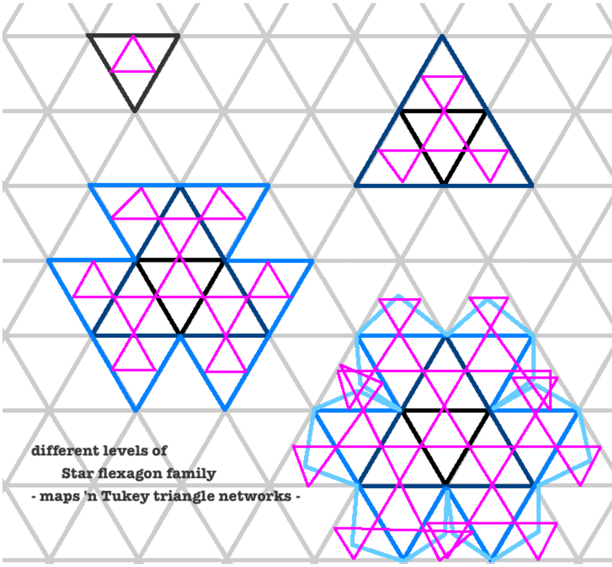

Fig. 1.29: StarFlexagonFamily-CharacteristicMapShape-difLevels

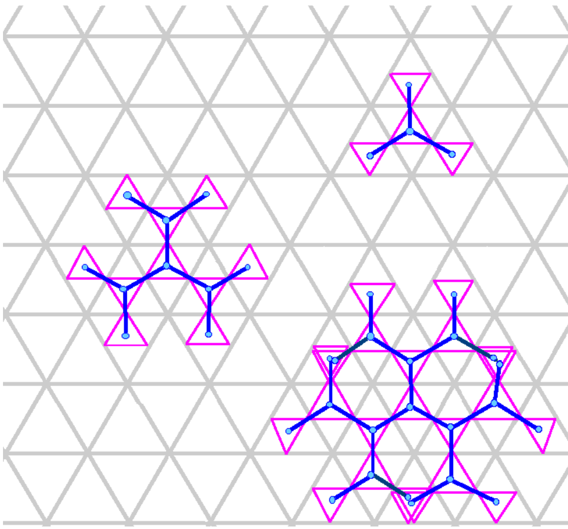

Fig. 1.30: StarFlexagonFamily-TukeyTriangleNetwork-TuckermanTree

Recursive behavior appreciated in the Tuckerman trees introduces a fractal conception in this context. Furthermore, each branch could be conceived as a pile-fractal. Doing a close up in a small Tuckerman tree's branch is easier to identify the fractal behavior which is nothing more than auto-reproduction/ recursive. Not forgetting the way in which this flexagon is constructed, one finds the pile-fractal. Star family nobly unveils these structures behind flexagons (fig. 1.31).

It could be said that two maps are equivalent if i) one of which is the other rotated around its center, ii) they are mirror images of each other. In other words if they are the same by rotation or

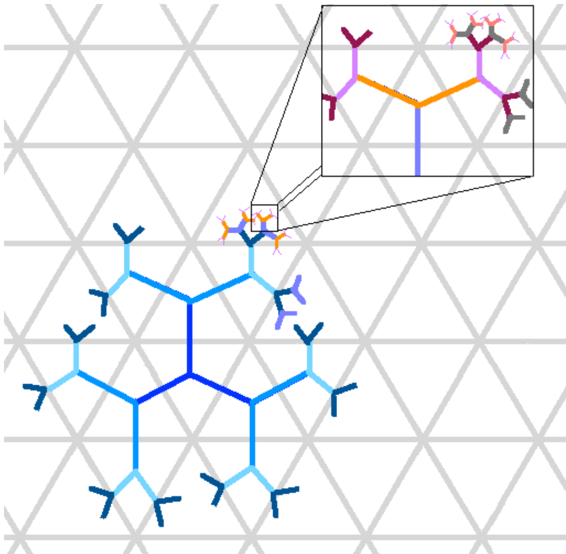

Fig. 1.31: StarFlexagonFamily-TuckermanTree

reflectiveness operations.

A flexagon is defined in terms of pats as an ordered pair of pats. Flexagons composed of different units, i.e. units that do not match; are named mixed flexagons. A cycle n flexagon still called a plain n-flexagon, is that which before returning to a specific side will execute n flexings.

In the folding process, the zero hinge is properly disposed such that two units of the flexagon are united by them. In that arrangement, the hinge in n position is called n-hinge.

Tubulation approximates a cube with the absence of a pair of opposite faces, i.e. a squared ring is obtained with its hinges out of its center. The absence of a consistent flexing (all left or right) along with no consecutive hinging defines an improper flexagon. A pair of tubulations are distinguished, hidden tubulations which result from attaching new sides, and exposed tubulations which arise from a sequence of flexes.

Hinge orientation derives from assigning direction to each side of the included polygons and with which hinges are still addressed. Indications to do that,

- By convention, the values increase in a clockwiseorientation, while hinge positions are disposed around the polygon.
- Draw arrows resting on the sides (both) of polygons -taking care of orientation-.
- Designate the zero-side.
- Finally, numbering following the vectors systematically.

Hinge difference is the arithmetic difference of the values assigned to a pair of sides of the same polygon where a couple of hinges were disposed. For convenience, the orientation which follows that of the vectors is designated as +; the contrary is called -. Such that a 2-hinge is +2 or ‡, a $(G - 1)$-hinge is -, and a $(G - 2)$-hinge is (=).

Absence of incomplete cycles will define a complete flexagon. A flexagon not derived from removing sides and having, at least partially, no consecutive hinging is recognized as an improper flexagon. Excluding at least one side by a shortcut denotes a cut which is classified according to its degree, the total sides excluded. Flexing carried on the contour of the polygon map is a 0-cut. The values of the implied flexes not mattering, a full series of 0-cuts will generate a 0-cut cycle.

Which is the same as going through the contour of the map.

The hinge difference across the tubulation must be the total differences of the 1-flexes crossed. Revising the class concept, the "number of sides in the constituent leaf polygons" is extended to include their orientation.

The degree of each face is defined as the number of hinge positions between the two hinges to each pat, measured about the center of the flexagon, plus one. The angle between the perpendiculars with respect to the arriving and leaving hinges is an alternative way to define the face degree. A face of degree d is denoted as d - *face*. (m, n) denote a face exhibiting side m above and side n below.

Rotation is defined as the order of the pat but inverted. 0^o-faces and 180^o-faces are equivalent. A group of flaps united on the course of a line at the flexagon center will generate a 0^o-face.

Dissimilar cycles (along with mixed plan polygons) assemble an heterocyclic flexagon. Combining two pats of distinct face degree will result in a compound face. Summing up the signs related to the leaves conforming the pat provides the the face degree of a pat. $(FR)^G = I$ defines the cycle G of a provided series of flexes.

1.3 Flexagon's properties related to concatenation operation

Here we have summed up some relevant notes [Conrad and Hartline, 1962] observed as properties during the study of extending a flexagon. "Extend a flexagon" means that to a given flexagon will be added some more sides. The process was tested on assembled as well as unassembled flexagons.

It was pointed out that in theory, following an inductive process it is conceivable no matter what flexagon, and that unexceptionally, the n-th flexagon will be produced from the preceding one, i.e. the n-th from the (n - 1)th. For such purpose, a technique of slitting along hinges was proposed and analyzed. A slitting process on the plan is equivalent to appending a new face on the map between two others which is part of the interdependence between the map and the plan.

In the case of maps, paths will be attached in pairs[20]. A pair of paths is the minimum number of them we can add together so that we start extending our flexagon. By the side of the plan, enlarging the flexagon using slitting and attaching operation is evident with the introduction of a pair of leaves in its structure.

In terms of signs, adding a side to a flexagon will consist of replacing some signs, for example, one plus is substituted for two minus (or one "-" for two "+") in the case triangles are added. But in general, the process of adding another cycle to a complete proper G-flexagon; consist of replacing a plus/minus sign by the opposite G - 1 minuses/pluses. Even more, given a flexagon sign sequence, opposite systematic reduction from several cycles to only one is feasible. This means to exchange G - 1 consecutive pluses or minuses by merely a contrary one. We give a couple of examples in figures 1.32 A, B and, 1.33 to illustrate this. A flexagon can be characterized by a reduction process of its sign sequence. The case of identifying a sign sequence with a flexagon is effortlessly solved in terms of a reduction like this.

Now, assuming an odd number of leaves. In the procedure of adding one more leaf, note that the number sequence of the subsequent unit will be reversed in reference to the first one. Bear in mind that, exclusively in end positions of the network lines will be allowed to enlarge the number of sides. And that a bijective relation is recognized between each flexagon side and each point along the contour of its map. Furthermore, an N - *gon* which is triangulated can be topologically equivalent to the map, i.e. the map of this flexagon may be represented by a polygon of N sides.

---

[20]P-Note: It should be clarified that in this context, path means the edge going from one vertex to another.

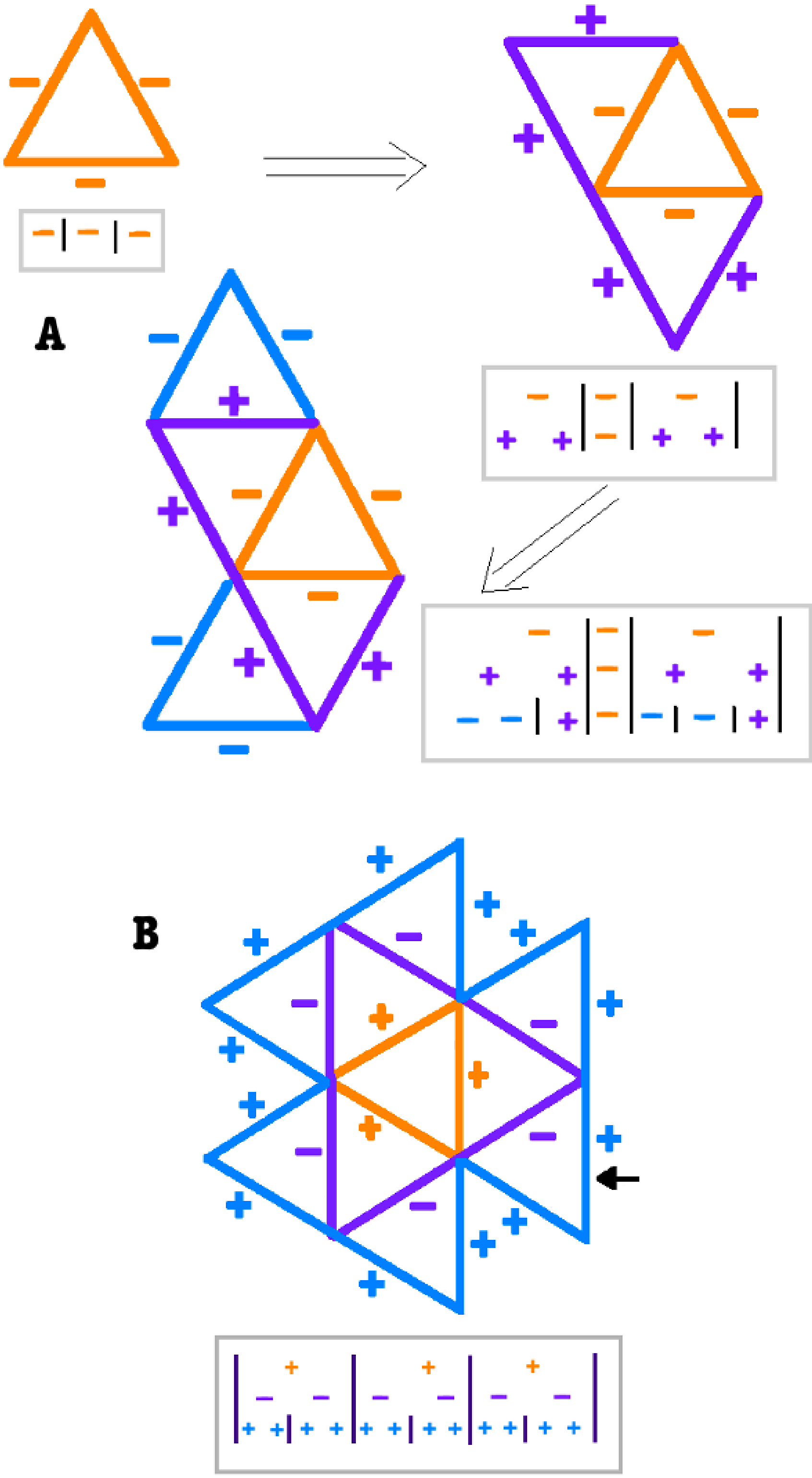

Fig. 1.32: ReductionSignSeq01-i123–Seq004 - step003

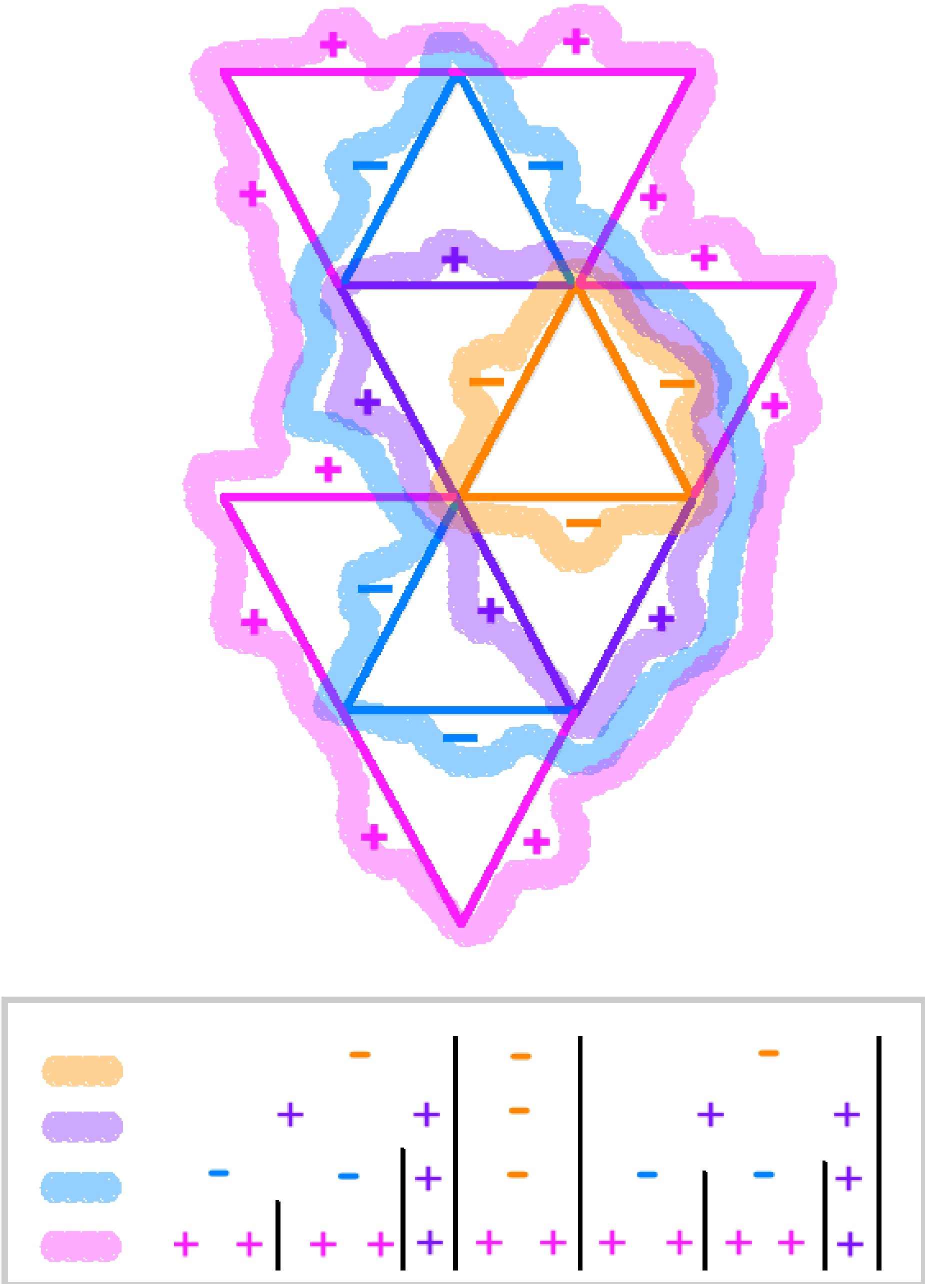

Fig. 1.33: ReductionSignSeq01-i4l

In accordance with more analysis, it had been noted that hypothetically, flexagons can be assembled with a base of any number of units. In addition, that in mixed flexagons each different unit has its own map however they hold a partial coincidence among all of them. Even more, it was anticipated that, hypothetically, a flexagon of any number of sides, units and of any polygon shape is conceivable. Moreover, it was clarified that no matter the polygon used, the Tukey Triangle System holds up.

The essence of the Polygon System comes from the fact that neighboring cycles have contrary orientation inverting plan signs[21]. However, it must be noted that the Polygon System will work fine merely for proper complete flexagons, i.e. a '+' or '-' addressing works aptly for proper flexagons however due to improper ones handle further than one, ± 1 it is not suitable.

It is observed experimentally that vectors in a new cycle added to a map will share the same orientation with last cycle on the enlarged side, whereas the rest will be opposite to it. In general, the vectors of cycles sharing one side will be opposite oriented relative to each other except the side in common. This reinforces the use of the Polygon System which facilitates the appreciation of this behavior at a glance.

Without delay, how the angles of the leaves are related to the angles of the map polygons acquires no significance which does not modify those operations used with regular leaves. What is important in the map is not the shape of the polygons involved, but the number of sides they have, i.e., as it was predicted, the cycle.

No matter which dimension of the second cycle is pursued it is attainable, just modifying the sort of the polygon incorporated into the manufacture of this cycle. The amalgam of cycles involved with no important, it is possible to manufacture a flexagon maneuvering several plans of polygons. Maps composed of mixed polygons are inherent to heterocyclic flexagons. The mechanism in which flexagon works is not influenced by the number of sides of the polygons utilized. Relative to this fact the class is irrelevant.

A substitution process exclusively relies on modifications happening in the number sequence which transform the flexagon's structure. While alterations in sign sequences affect both, the arrangement of hinges and the contour of the leaves. Thus with only one necessary sign sequence restriction remaining, the shape of the flexagon can get an unlimited variety.

According to a Pook [Pook, 2009] analysis, he identified four main position in which flexagons can be joined. They are called flat, box, skew and, slant main position links.

[21] The corresponding fragment of the flexagon sign sequence.

Chapter 2

Correlating to graphs

The current chapter submits what we need from graph theory in order to connect it to the context of flexagons. We know that König did not standardize the nomenclature to graphs the time he formalized graph theory in his book [Konig, 1990], which is a basic reference for it, so the familiar terms alluded in the graph context are used here. After some definitions, it can be comprehended how a drawing concentrates large information about how their objects are related. In addition to understanding how convenient this type of representation is.

The work of Hassler Whitney [Whitney, 1931] was the base to present dual concept, which clearly shows the dual relationship. Parallel of getting basic definitions, the convenient relations to flexagon theory are submitted. An assortment of figures shows step by step how to get some conceptions in order to clarify and illustrate each given notion. The perspective of König [Konig, 1990], Whitney [Whitney, 1931], Béla Andrásfai [Andrásfai, 1977] along with the authors cited in the current chapter is reflected in the supplementary explanations adding clarification to the introduced concepts.

2.1 Short in graphs

2.1.1 Graph definition

In this document, to represent the objects in our drawing as an alternative to simple points, the idea of using small circles is adopted. Basically, to avoid confusion with other points[1] generated from the sketching process. Those small circles will represent the vertices. The edges will be the lines connecting them in pairs representing the existing relationship between them.

The vertices at the end of each edge are called endpoints and, next lines will describe how they are associated. The endpoints of an edge are said to be adjacent to each other. An edge is said to be incident to both of its endpoints. The possibility of finding (naturally called) isolated vertices is open[2]. How many edges are incidental to one vertex is said to be the degree[3] of that vertex. After some basics given above, it is possible to supply a definition of a graph, which was taken from [Bondy and Murty, 2008].

Definition 2.1.1. A graph G is an ordered pair $(V(G), E(G))$ consisting of a set $V(G)$ of vertices and a set $E(G)$, disjoint from $V(G)$, of edges, together with an incident function ψ_G that associates with each edge of G an unordered pair of (not necessarily) distinct vertices of G.

That is, a drawing whose components are vertices and edges where the latter represents how the former are related. $G = (V, E)$ is a frequent notation used whose set of vertices is represented by V and E denotes the set of edges in the graph G. We should have thought of the edges as the part of the graphs condensing information of how the vertices are correlated with each other. In fact some authors use a trio (V, E, R), in place of a pair, setting apart the set of relations among the vertices from the edges' set.

2.1.2 Relation between graphs

Briefly, some concepts about how graphs are related to each other are included. Given a couple of graphs G_1 and G_2, if it is possible to set up a unique correlation of vertices with vertices and edges with edges such that their adjacency property is held, then G_1 is said to be homeomorphic to G_2 and vice versa. A complete graph is such that each vertex is joined to each other by an edge. Given any graph G, the complement is defined based on edges only, by the set of needed edges to have a complete graph. In other words [Harary, 1969],

Definition 2.1.2. The complement $\bar{G}$ of a graph G also has $V(G)$ as its point set, but two points are adjacent in $\bar{G}$ if and only if they are no adjacent in G.

Note that this is different from the subgraph concept in which both sets, the vertices and the edges are involved. To obtain a subgraph given a graph G, some edges and their corresponding incident vertices were removed. If the subgraph is not the same as the original one, then we have a proper subgraph. Defined in [Gibbons, 1985],

Definition 2.1.3. A (proper) subgraph of G is a graph obtainable by the removal of a (non-zero) number of edges and/or vertices of G. The removal of a vertex necessarily implies the removal of every edge incident with it, whereas the removal of an edge does not remove a vertex although it may result in one (or even two) isolated vertex.

In the succeeding lines, the previous information will be complemented with some concepts revealing how vertices are related to each other in pairs. If a couple of vertices are arbitrarily chosen such that it is possible to set up in a certain way, think of it as a road, in order to go

[1]Ex. intersection points.
[2]In flexagons it makes sense when physically it is hard to visit all faces – although in theory it is possible –.
[3]Also named valency.

through the graph from one vertex to the other. Where visiting the other vertices is allowed only once. If moreover, by this "way" is free of -inner-[4] cycles, that way is defined as a path. The first and the last vertices of it are called the endpoints of the path. Intermediate points are said to be its inner points. Strictly defined by [Diestel, 2000]

Definition 2.1.4. A path is a non-empty graph $P = (V, E)$ of the form

$$V = \{x_0, x_1, \ldots, x_k\} \qquad E = \{x_0x_1, x_1x_2, \ldots, x_{k-1}x_k\}$$

where x_i are all distinct. The vertices x_0 and x_k are linked by P and are called its ends; the vertices $x_1, \ldots, x_{k-1}$ are the inner vertices of P.

A path is a walk conditioned to have all its vertices distinct. Let us define it [Diestel, 2000],

Definition 2.1.5. A walk (of length k) in a graph G is a non-empty alternating sequence $v_0e_0v_1e_1 \ldots e_{k-1}e_k$ of vertices and edges in G such that $e_i = \{v_iv_{i+1}\}$ for all $i \leq k$. If a $v_0 = v_k$ the walk is closed.

Length definition and some other concepts will close this section. How many edges belong to the path defines its length. In the case of a closed path, we have a circuit. When fixing one vertex the concept of length still applies to it. A connected graph is said to be a graph where every pair of vertices can be connected by a path. If the graph is not connected, it is said to be disconnected. Finally, a connected subgraph is defined as a component.

2.1.3 Universal Flexagon in Terms of Graphs

Conrad and Hartline characterized the flexagons as a space, fig 2.1 illustrates[5] the covering space [Conrad and Hartline, 1962] introduced and explained by them which represents what they called the ultimate flexagon[6]. From which whatever desired or requested flexagon can be obtained by conveniently removing the edges needed in order to attain it. Let us show an example in figure 2.2 where we still see the correspondence between a subspace of the covering space and the given flexagon map.

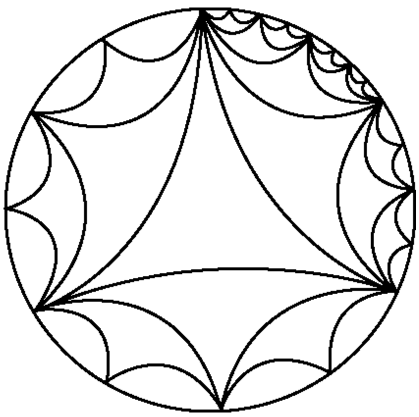

Fig. 2.1: Covering Space to represent flexagon's maps

[4] The whole way can be a cycle.
[5] Original borrowed from [Conrad and Hartline, 1962]
[6] Here referred to as the Universal Flexagon.

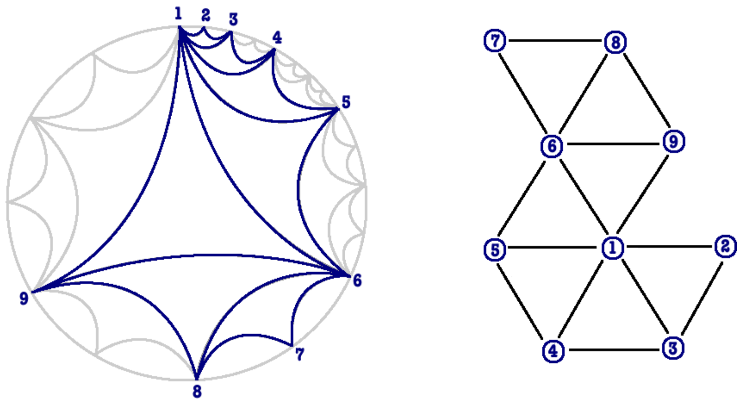

Fig. 2.2: Covering Space to represent flexagon's maps, example

From here we can achieve the Polygon System. Let us associate the set of vertices with the (finitely many) set of natural numbers (i.e. $V \subseteq N$). Arranging the vertices around the circle such that they coincide with the vertices of a n-gon, then we replicate connections among them exactly as in the covering space. Figure 2.3 shows us the proposed subspace (right) and the new arrangement (left) corresponding to the example above.

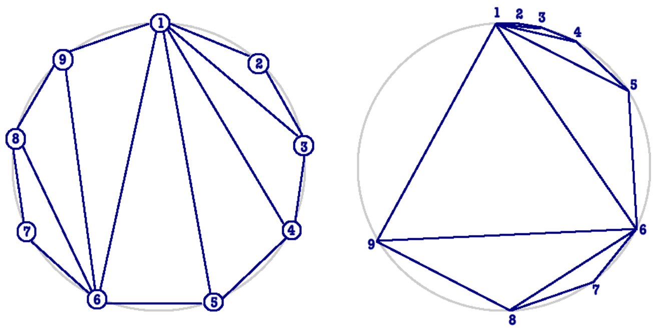

Fig. 2.3: Subspace-Polygon System

Which is a proper subgraph of the G_{UF}[7]. With this fact, it is easy to appreciate that whatever flexagon can be obtained from the G_{UF} - universal flexagon. Figures 2.4, 2.5 and 2.6 represent one more example. Furthermore, let us exemplify a couple of different possible frieze (patterns | layout | plan) obtained from the crocodile map. Figures 2.7 and 2.8 show full information. Notice that figure 2.8 is adding the specifications of the suggested reduced sign sequence which validates that the initial map can produce a flexagon.

2.1.4 Dual

At this point, it is followed the approach of Hassler Whitney [Whitney, 1931] to present the dual concept. This section is started by jumping into an example. Our initial graph is the propeller in illustration 2.9.A. After inserting a second graph the picture will look like that in fig. 2.9.B, now we proceed to dot it in the way showed in fig. 2.9.C, notice the dot resting out of the border.

[7] The universal flexagon graph.

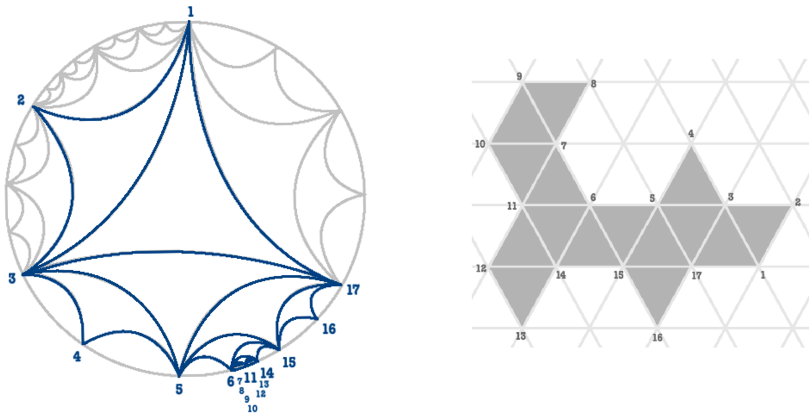

Fig. 2.4: Covering Space - subspace crocodile

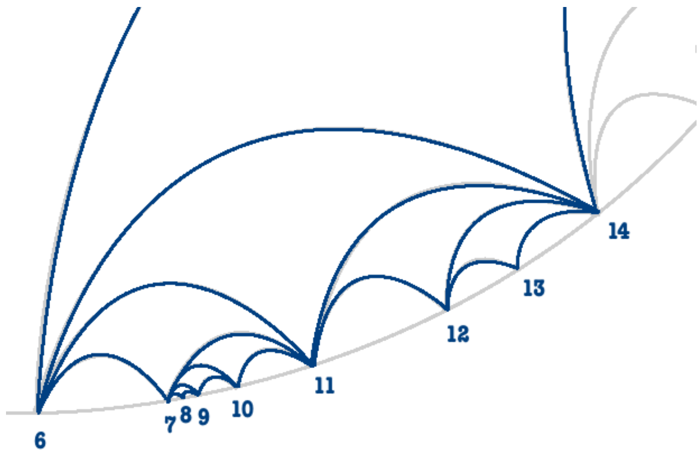

Fig. 2.5: Covering Space - subspace crocodile-zoom-in-v-14-6

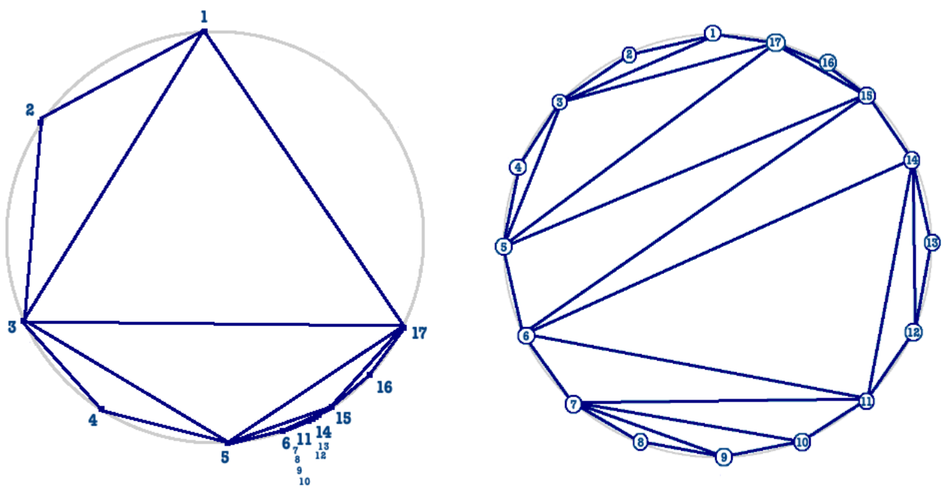

Fig. 2.6: CS-coco

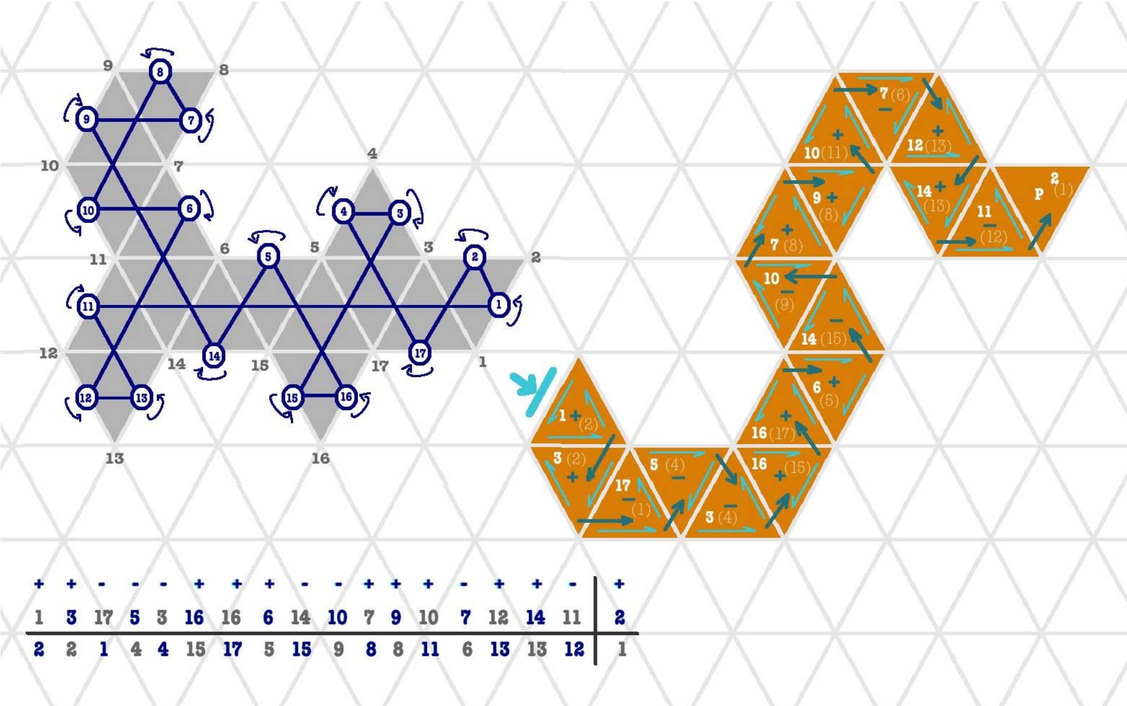

Fig. 2.7: FC-map-pat-coco-01

Clearing slightly the illustration, in order to keep an intelligible representation to then complete one more graph, the resulting dots (small circles) in the dotting process will be the new vertices and connecting them as it is shown in fig. 2.9.D, will give us the edges.

Before we continue, one more relevant definition will be introduced, the planar graph concept. Given a graph G, if it is possible to get one from it, a representation on the plane or on a sphere in such a way that none of their edges crosses them, such graph a planar graph. [Behzad and Chartrand, 1971] formalize the notion, not before clarifying what embeddable means as part of it.

Definition 2.1.6. A (p, q) graph G is said to be realizable or embeddable on a surface S if it is possible to associate a collection of p distinct points on S which correspond to the vertices of G and a collection of q Jordan arcs, mutually disjoint except possibly for endpoints, on S which correspond to the edges of G such that if an arc a correspond to the edge $e = uv$, then only the endpoints of a correspond to the vertices of G, namely u and v.

Definition 2.1.7. A graph is planar if it can be embedded in a plane (or, equivalently, on a sphere).

One thing to remark is that a planar graph accommodated on a sphere helps us to distinguish how it divides the sphere into regions (fig. 2.10.A). Observe that all outside the border of the graph is forming R_a, the region a.

The procedure to build the dual of the initial graph G is to dot each region then, to join the new vertices, the internal ones having sides on the border with the vertices in the region outside the border of G and, the internal ones are joined in the natural way they are presented. The new sides are restricted to cross the old ones only once. Already illustrated above on the plane, now let us appreciate it on the sphere (fig. 2.10.B). The dual graph is defined in the "Graph Drawing" book [Batista et al., 1999] as:

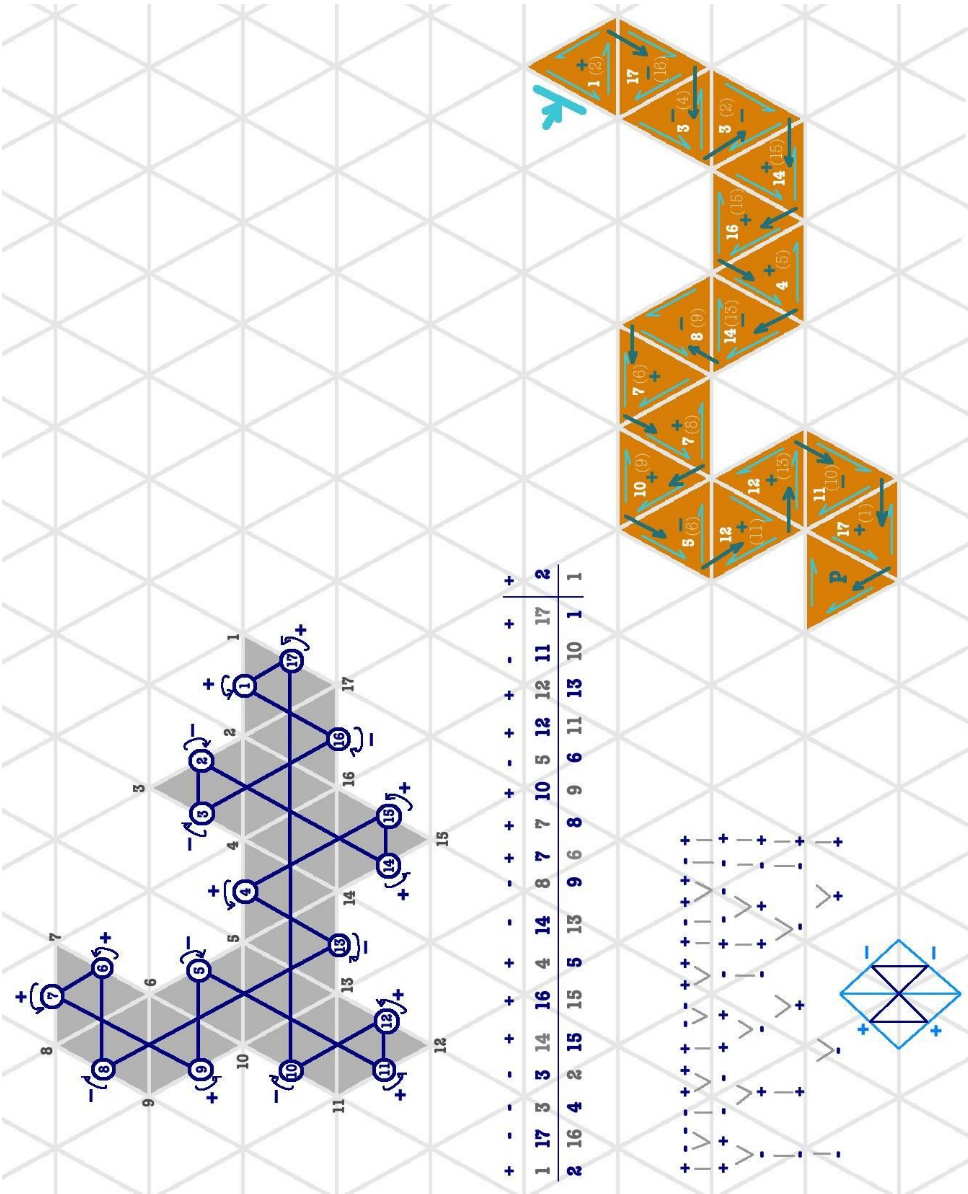

Fig. 2.8: FC-map-pat-coco-02

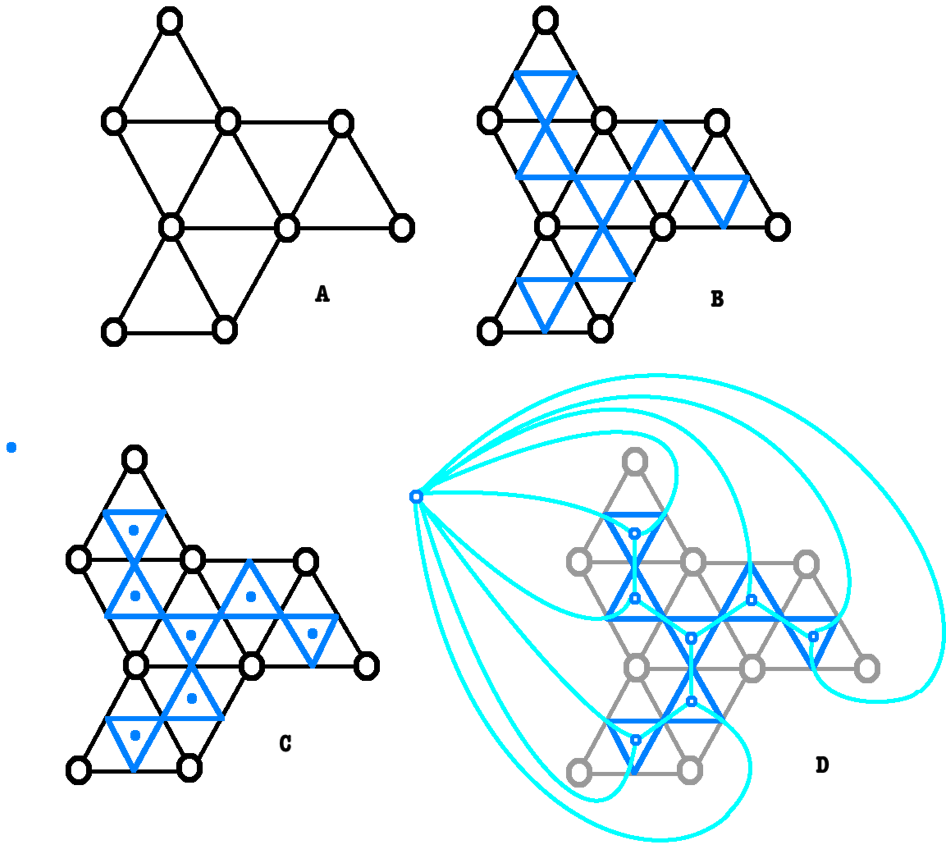

Fig. 2.9: Propeller-Map-TTN-dual

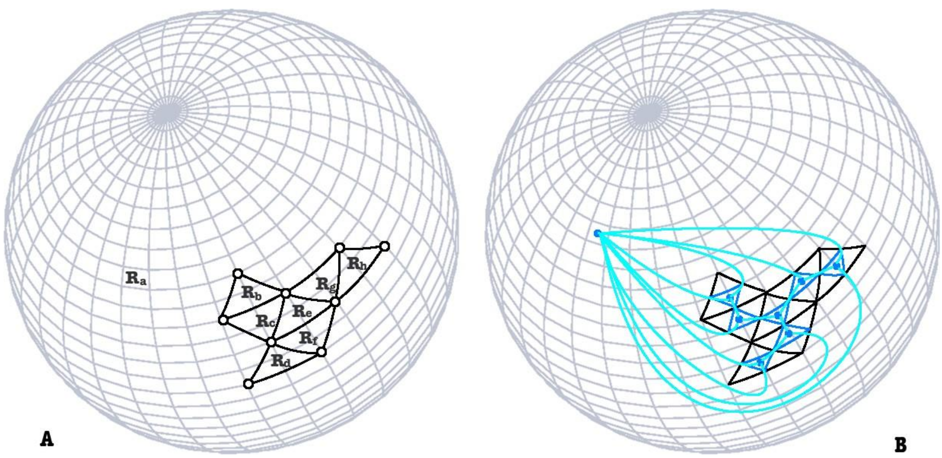

Fig. 2.10: Sphere-sphere-graph-regions | graph-dual

Definition 2.1.8. The Dual Graph G^* of an embedding of a planar graph G has a vertex for each face[8] of G, and an edge (f, g) between two faces f and g for each edge that is shared by f and g.

Understanding the embedded graphs as the different possibilities in which, holding their correlations, vertices and edges can be arranged. Furthermore, observe that due to the restriction on the triangulated surface to be a flexagon map and the process of normalization, there is no variation in such distribution then the embedded graph is the given graph.

2.1.5 Trees and forests

The essential concepts about trees will be introduced to prepare us to associate them with flexagons. Let us start with the basic concept of a tree which is defined as a connected graph with no cycles. The unique request to have a forest is the absence of cycles in a given graph, notice that the latter description is not conditioned to be a connected graph.

Based on fig. 2.10.B, let us remove all sides incident to the vertex in the external region, the tree resultant is what we know as the Tuckerman Tree. Then, the Tuckerman three is naturally defined as a subgraph of the dual of the graph which represents the map of a flexagon. Picture 2.11 illustrates that.

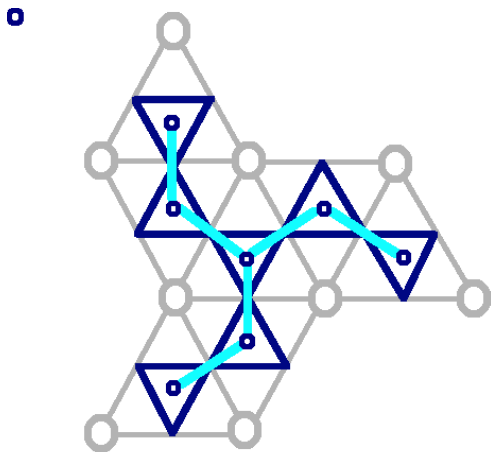

Fig. 2.11: Tuckerman-Tree-SG

Given a connected graph the simple way to obtain a spanning tree is defining a subgraph, holding all vertices of the original graph and removing necessary sides such that we get a tree. Considering that a subgraph of a graph G spans G if it contains every vertex of G. Formally it is defined by [Wallis, 2007],

Definition 2.1.9. A spanning tree is a spanning subgraph that is a tree when considered as a graph in its own right.

The **dual of the Tukey Triangle Network**, the Tuckerman tree is naturally defined as a spanning tree. Actually, as a kind of spanning tree due to all sides connecting with the vertex in the external region where removed from the flexagon map's dual, we are solely keeping such sides connecting internal triangles into the flexagon map. In fact, giving a map of a flexagon the Tuckerman tree is strictly a subgraph of its spanning tree. The set of figures 2.12.A, B, C, D. E and, F illustrates that.

[8] Region.

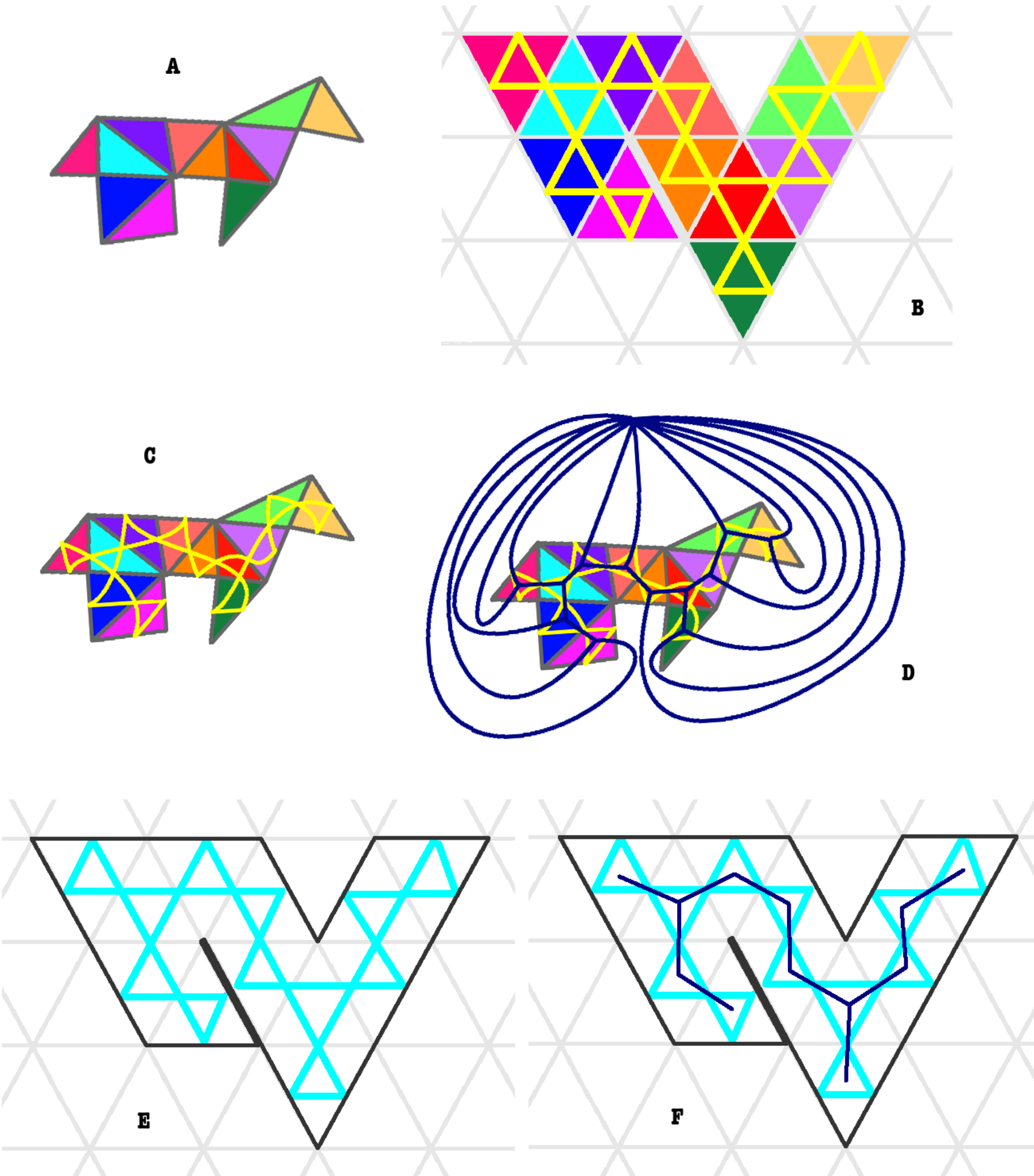

Fig. 2.12: Horse-triangled-base-ttn-dual-ttree

Given a connected graph G, the fundamental system regarding a spanning tree is the set of edges whose elements have two special attributes:

- each edge is part of a 1-cycle/circuit,
- each 1-cycle has, as one of its edges,
  one and only one of the edges removed in the process of getting a spanning tree.

Figs 2.13.A, B, C constitute the first example. And figs 2.14.A, B, C, D, E, the second one. The union of the spanning trees components [9] will be called a spanning forest.

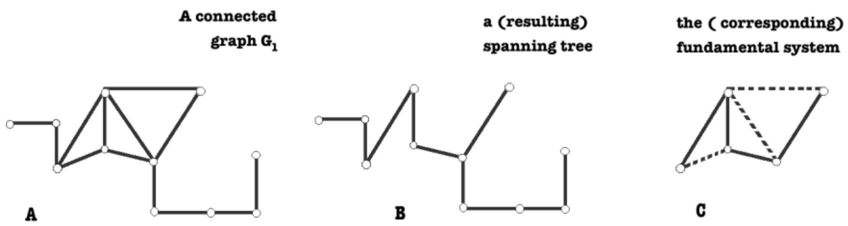

Fig. 2.13: Example1 - ConnectedGraph-ST - the fundamental system (base on ST)

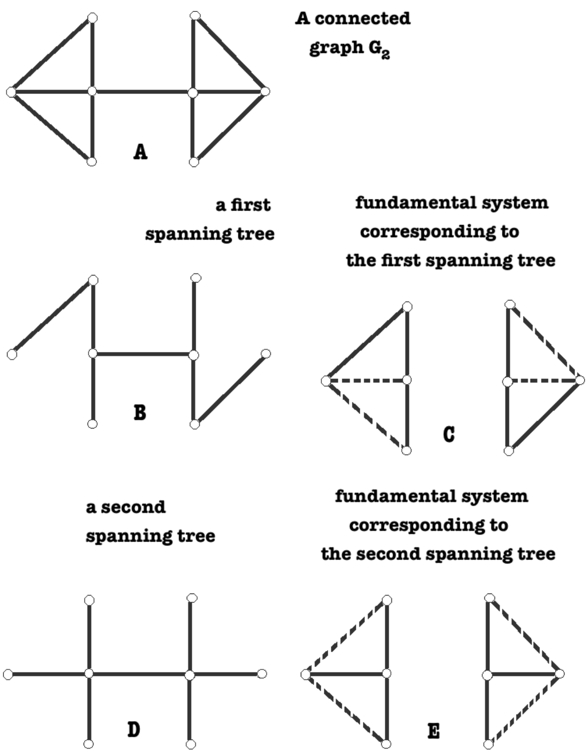

Fig. 2.14: Ex2 - ConnectedGraph-ST1 | ST2 - the fundamental systems (base on ST1 | ST2)

If Tt denotes "Tuckerman tree", St a spanning tree and, A the set of sides having to a as an

[9] One spanning tree for each component.

endpoint where $a \in$ dual is the vertex in the external region. We can say that

$$Tt \cup \epsilon_a = St \quad \text{where } \epsilon_a = \epsilon_i \in A.$$

2.1.6 Directed graphs

Once an orientation is assigned to each edge of the graph then, a directed graph[10] is obtained. It is also named as a "digraph". Formally, [Bondy and Murty, 2008],

Definition 2.1.10. a directed graph D is an ordered pair $(V(D), A(D))$ consisting of a set $V := V(D)$ of vertices and a set $A := A(D)$, disjoint from $V(D)$, of arcs, together with an incidence function ψ_D that associates with each arc of D an ordered pair of (not necessarily distinct) vertices of D.

In the case of the arc concept, [Wilson, 1985] helps us with its definition,

Definition 2.1.11. An arc whose first element is v and whose second element is w is called and arc from v to w and is written (v, w) or simple vw.

we will sum up some properties in reference to them. Given the sequence of edges

$$e_1, e_2, e_3, \ldots, e_{r-1}, e_r, \ldots, e_n,$$

where the endpoint of e_{r-1} is the initial point of e_r, such sequence will denote an edge train. This definition is valid to both directed and no directed graphs.

Henceforth, the term oriented or directed will be used indistinctly. Given an oriented graph $\vec{G}$ when a defined edge-train contains all edges of it, the graph $\vec{G}$ will be called an Eulerian line; and it can be opened or closed, i.e., when the endpoint of e_n is the initial point of e_1. A directed graph can be called, either a directed path if it is an open Eulerian line and also holds a path once each orientation is removed; or a directed circuit if it is a closed Eulerian line and yet holds a circuit once all orientations were taken off.

The Tuckerman Traverse is, as might be expected, defined as an oriented circuit. To generate the frieze code, we already explained that we collect information according to the structure following the established addressing. Said structure is the Tuckerman Traverse and the data collected are the vertices' tags attributed to it. Remember we assigned a positive sense to the flexagon map, the base for the Tuckerman traverse. This is the same '+' orientation in which we start going through the Tuckerman Traverse.

Some authors do not differentiate the Tukey Triangle Network from Tuckerman Traverse. Let us see how we understand them. We can say that the structure obtained according to the rule of joining each middle side vertex, the sides on the border of the flexagon map in such a way that they are restricted to reflect in an angle less than a right angle until we get the initial point again. This structure is called the Tukey Triangle Network and is one dual of the flexagon map. The manner that such structure is traversed is what is named the Tuckerman Traverse. If we get a map of traversing the flexagon, the obtained structure could be called the Tukey n-gon Network and the go through will be the Tuckerman traverse. In terms of oriented or non-oriented graphs, we could say that the Tuckerman Traverse is an oriented graph and that the Tukey n-gon Network is its corresponding non-oriented graph.

Let us see it by examples. We start with the flexagon map, the shadowed area in the fig. 2.15.A then, we complete the Tukey Triangle[11] Network (fig. 2.15.B). We clear up the picture to leave the Tukey Triangle Network alone (fig. 2.15.C). We are ready to add orientation thus getting the Tuckerman Traverse (fig. 2.15.D). A circuit going through all vertices of a given graph is said to be a Hamiltonian circuit. In the case of a path accomplishing the same property, it will be

[10] Referred as oriented graph in this document.
[11] In this case.

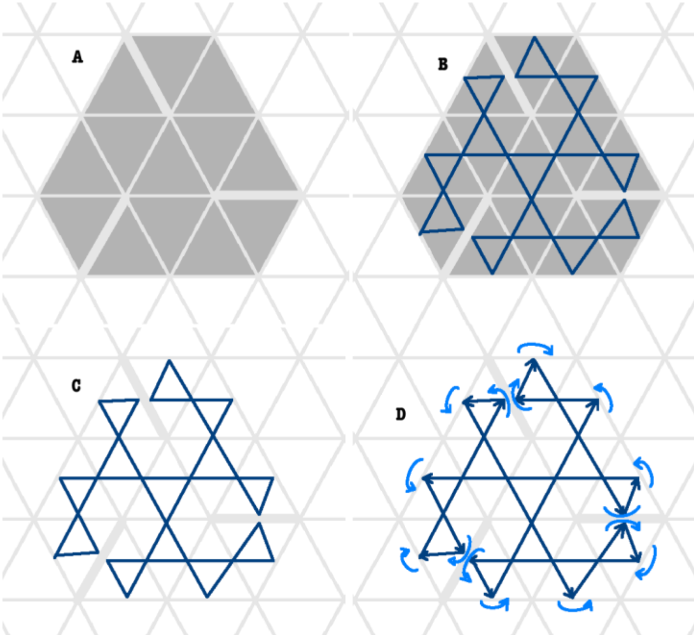

Fig. 2.15: Flexagon's map-TTN | unoriented 'n oriented graph

named a Hamiltonian path.

Let us see what happens in **The Tukey Triangle Network**. The Tukey Triangle Network is defined in base on the flexagon map touching each external side of its border in such a manner that we touch each one until we achieve the starting side. We can see that Tukey Triangle Network is, by nature, conceived as a circuit. Furthermore, each side is touched only once, then it can be said that the Tukey Triangle Network is a Hamiltonian circuit.

One more example, think of the shadowed region in fig. 2.16.A as the map of our flexagon. Then we draw the line representing the Tukey Triangle Network. Full data is concentrated in fig. 2.16.B. Making this slightly clearer, the corresponding Tukey Triangle Network was placed alone in fig. 2.16.C.

2.2 Matrices

Matrices are inheritable related to graph theory, strictly the essentials will be introduced. Enough to show what the characteristic matrices of the graphs related to flexagons could be.

2.2.1 Adjacency

When the value of the element a_{ij} shows how many times v_i and v_j are connected, where v_i, v_j ϵV which is the set of vertices of a given graph G then the matrix $A = [a_{ij}]$ becomes the adjacency matrix[12]. Formally [Bollobás, 2000],

Definition 2.2.1. The adjacency matrix $A = A(G) = a_{ij}$ of a graph $G = (V, E)$ is the nxn matrix given by:

$$a_{i,j} = \begin{cases} 1 & \text{if } v_i, v_j \quad E(G) \\ 0 & \text{otherwise} \end{cases}$$

[12]The adjacency matrix tell us if two vertices are connected and that's it.

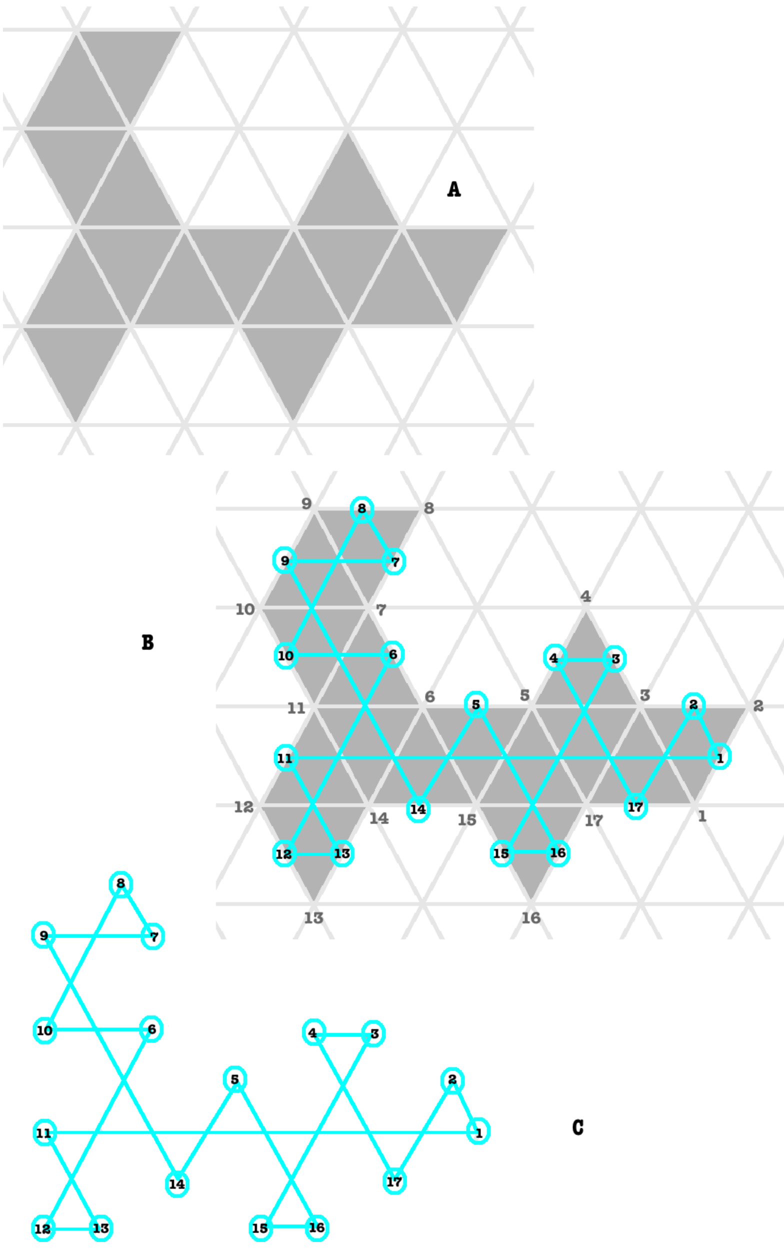

Fig. 2.16: Map's flexagon-cocodrile-ttn-lnv-lnva

A symmetric matrix is obtained if the G is undirected otherwise an asymmetric one is generally attained. In the latter, the value of a_{ij} is counted if the edge is oriented from v_i to v_j.

It should be noted that the adjacency matrix is a squared matrix and its order will be determined by the number of vertices we have in V. Example, given the graph in fig 2.17, which corresponds to a tetra-flexagon map.

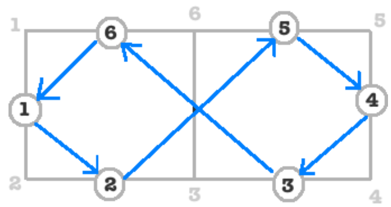

Fig. 2.17: Adjacency-matrix-ex-bi-tetra

Next values were obtained,

$$V = \{1, 2, \ldots, 6\}$$
$$O_A = 6$$

$$[a_{ij}] = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Where V is the set of vertices, O_A is the order of adjacency, and $[a_{ij}]$ is the adjacency matrix. Now, examining the isomorphic property between a pair of graphs. Once both graphs are conveniently denoted, matrix notation could simplify the process to determine if the given graphs are isomorphic to one another. In the equation $A_1 = P^{-1}A_2P$, the permutation matrix is denoted by P and A_1, A_2 are the adjacency matrices corresponding to the given graphs. Such equality means that the given graphs are isomorphic always P exists.

2.2.2 Incidence

The incidence matrix is considered the edge matrix because the elements b_{ij} describes the manner by which the edges and vertices structuring the given graph are placed. Not considering the loops, but observing an orientation of the edges. The concept is established by [Bollobás, 2000]:

Definition 2.2.2. The incident matrix $B = B(G) = b_{ij}$ of G is the nxm matrix defined by

$$b_{ij} = \begin{cases} 1 \text{ if } v_i \text{ is the initial vertex of the edge } e_j, \\ -1 \text{ if } v_i \text{ is the terminal vertex of the edge } e_j \\ 0 \quad \text{otherwise} \end{cases}$$

In the case of unoriented graphs, incidence is the only prevalent condition and 1, 0 are the valid values. To the incidence matrix, the index of the rows corresponds to vertices and the one of columns is associated to edges.

Incidence matrix is also useful to determine an isomorphic relationship. Given two graphs (oriented or not), with no loops, are said to be isomorphic if the incidence matrix can be deduced from one another just adjusting the order of rows and columns.

2.2.3 Circuit

For the circuit matrix the valid values will be {0, -1, 1} if the involved graph is oriented and, {0, 1} if not. Here the values indexing the circuits will correspond to the indices of the rows and the edges to the columns.

$$k_{ij} = \begin{cases} 1 \text{ if the edge } e_j \text{ belongs to the circuit, with same orientation, to } k_i \\ -1 \text{ as above, but contrary orientation} \\ 0 \text{ otherwise} \end{cases}$$

2.2.4 Tukey Triangle Network

In some sections above the concept of Tukey Triangle Network has been mentioned, here we will be more familiar with it. Originally, it was defined as a map formed by triangles then, let us start in that way. Given a graph representing a flexagon map whose primary polygons are triangles; we will define an homeomorphic[13] graph to it. For this purpose, we will establish each new vertex on the half of each original side and let us set the minor value between each original pair of vertices as the value of the new vertex (fig. 2.18.A). We will link them going from one vertex to the next as if we were to imitate the reflection of a light ray on a mirror (fig. 2.18.B). Figures 2.18.C and 2.18.D illustrate the complete new graph in blue.

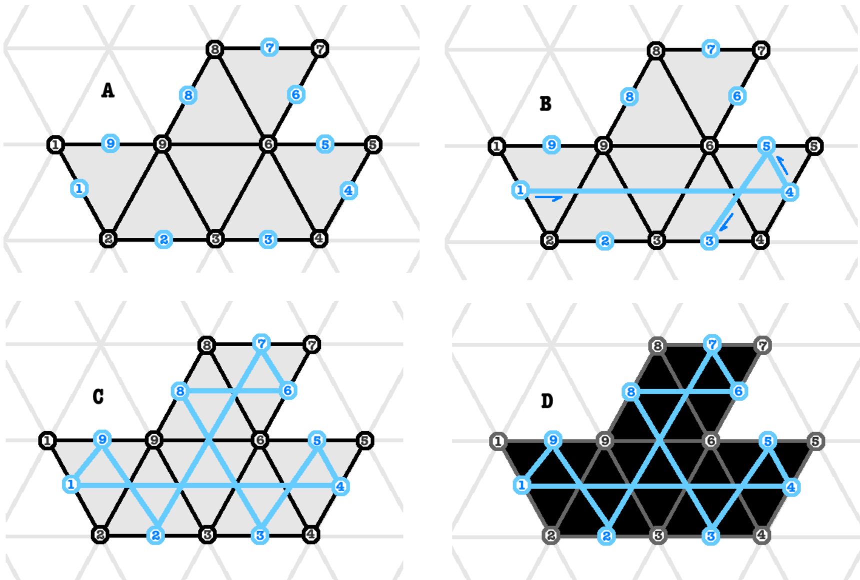

Fig. 2.18: Triangled-base-map R | TTN | irttna

At this point, let us introduce the fundamental theorem, the theorem I presented by Hassler Whitney in his paper "a theorem of graphs [Whitney, 1931]" copied right below. For the proof's detail the [Whitney, 1931] can be referred to.

Theorem 2.2.1. *"Given a planar graph composed of elementary triangles, in which there are no circuits of 1, 2, or 3 edges other than these elementary triangles, there exists a circuit which passes through every vertex of the graph."*

Despite the Tukey Triangle Network is a circuit which visits every vertex only once, in fact, it is the flexagon map from it was derived which seems to fulfill such theorem.

2.2.5 Normal Form

Hassler Whitney described the the normal form [Whitney, 1931] as the construction of a circuit satisfying that it touches each vertex of a planar graph only once which is strictly formed by triangular regions. The process consists of arranging the nodes such that they are the vertices of a n-gon and adjusting the sides, shorter or longer attending planarity as well as avoiding crossing among them, holding the relations they had in the original graph. The polygonal form of a flexagon map according to the Whitney definition is the normal form of that flexagon map.

[13] In the sense of that the obtained graph preserves similarity to the first one.

The normalization process can be applied to any graph satisfying the required conditions on it. The graphs representing a flexagon map always fulfill them, McIntosh worked normalizing them directly rather than one of its duals. In fact after having normalized each map proposed for the joining and attaching of them, a normalizing process operates once more, now on the whole result. Only after that, a dual is obtained which allows us to collect the joining number sequence easier.

Chapter 3

Grammars, Flexagons and Catalan Numbers

In the present chapter, it is supported that Catalan Numbers solve the question of how many flexagons can be gotten given a polygon from which a map is obtained for each possible polygon partition. Pointing out that the current document is working on joining flexagon maps via their corresponding normalized graphs. For such purpose, the basis to get immersed in the concepts related to grammar are presented. Certainly that, they can be thought of much the same as those to recognize the notions applied in the process of producing a natural language, the context is into the computer theory. A summary of the perspective of Vun Belcher is presented solving the problem of sectioning a surface in triangles which has been treated from different perspectives depending on the purpose for which it is being studied. He observed that he derived the Catalan Numbers following his own deductions. An outline of the grammar that Lentin and Gross defined to solve the same problem is introduced, along with a refresh of the concept of *Z*-side to couple with the virtual side corresponding to the starting state of that Polish grammar. Once we have the concepts together, they are connected to justify the correspondence with Catalan Numbers.

3.1 Preliminar definitions

Generally, we understand a grammar as a set of symbols obeying certain conventions to get meaning. In short, the grammar produces a language. One method to specify a grammar uses four elements: a terminal set, a nonterminal set also referred to as variables, its start state and, the productions set also named rules or transition functions. The terminal and nonterminal sets are a pair of disjoint finite sets of symbols.

According to the particular purpose of study an alphabet is set, the terminal symbols set over which the language is built. The ASCII characters set, Roman or Arabic numbers or even the Greek alphabet are all valid candidates to declare them as alphabets. Or we can even define it as the binary set {0, 1}.

Nonterminal symbols are subject to replacement obeying the rules defined. The productions are constituted of three components, on the left by at least one single nonterminal or one nonterminal prefixed and/or suffixed by the combination of zero or more terminal and nonterminal symbols, the → symbol in the middle and, a combination of zero or more terminal or nonterminal symbols on the right. Simply phrased, they are regulating the formation of the words belonging to our language.

That is to say, the set of words produced from the alphabet according to the transition function will be referred to as the language. The directives which allow us to operate the generated language are named operators. The commonly recognized operators are union, concatenation, star (or Kleene closure).

Formally a grammar [V.Aho and Ullman., 1972] is a 4-tuple where each element indicates the ones explained above in the same order and denoted by $G = (N, \Sigma, P, S)$.

Definition 3.1.1. A grammar is a 4-tuple $G = (N, \Sigma, P, S)$ where

i) N is the finite set of nonterminal symbols, also termed variables.
ii) The greek letter Σ denotes the finite set of terminal symbols.
iii) The productions, think of them as the generating rules. Strictly, they are a subset of $(N\ U\ \Sigma)^* N (N\ U\ \Sigma)^* \times (N\ U\ \Sigma)^*$.
iv) The S symbol denotes the starting symbol indicating the initial state.

To better understand, let us define the next example. Let it be

$S \to aSS,$
$S \to b$

P, the set of transition functions and the grammar will be defined by $G(\{S\}, \{a, b\}, P, \{S\})$.

Obeying the *Chomsky hierarchy*, among its grammars' classification are the right-linear, the context-free, the context-sensitive as well as the unrestricted [V.Aho and Ullman., 1972]. They are defined according to the style of their productions. i) $A \to xB\,|\,x$ where A, B are non-terminals and $x \in \Sigma^*$ are productions which characterize the right-linear grammars. ii) $A \to \alpha$ where $A \in N$ and $\alpha \in (N \cup \Sigma)^*$ are the ones related to the context-free grammars. iii) The context-sensitive grammars are those whose productions have the form $\alpha \to \beta$ where $|\alpha| < |\beta|$. The unrestricted grammar has no restrictions as those obeying the other three. Notice that the right-linear grammars are a subset of the context-free grammars. A right, or left in case the rule production is of the form $A \to Bx\,|x$, grammar is called a regular grammar[John E. Hopcroft, 1979].

Having a grammar in a one-line form with the root as the only nonterminal and exclusively using operators and terminals in their rules productions, such grammar is called a regular expression. It consists just of terminal characters, parentheses for grouping, and operator characters [Amarasinghe et al., 2015]. A recursive definition [V.Aho and Ullman., 1972] of Regular expressions over Σ and the regular sets they represent is given by

Definition 3.1.2. (1) $\emptyset$ is a regular expression denoting the regular set $\emptyset$.

(2) e is a regular expression denoting the regular set $\{e\}$.
(3) $a \in \sum$ is a regular expression denoting the regular set a .
(4) If p and q are regular expressions denoting the regular sets P and Q, respectively, as follows

   (i) $(p + q)$ is a regular expression denoting $P \cup Q$.
   (ii) (pq) is a regular expression denoting PQ.
   (iii) $(p)^*$ is a regular expression denoting P^*.

(5) Nothing else is a regular expression.

p^+ will denote pp^*, and parentheses will be removed always than ambiguity does not arise.

Carroll and Long [Carroll and Long, 1989] define separately regular sets and regular expressions which helps to better comprehend their recursive definition 3.1.2.

Definition 3.1.3. Let S = $\{a_1, a_2, \ldots, a_m\}$ be an alphabet. A regular set over $\sum$ is any set that can be formed by a sequence of applications of the following rules:

i. $\{a_1\}$, $\{a_2\}$,......, $\{a_m\}$ are regular sets.
ii. {} (the empty set of words) is a regular set.
iii. **{λ}** (the set containing only the empty word) is a regular set.
iv. If L_1 and L_2 are regular sets, then so is $L_1 \cdot L_2$.
v. If L_1 and L_2 are regular sets, then so is $L_1 \cup L_2$.
vi. If L_1 is a regular set, then so is L^*_1.

Definition 3.1.4. Let S = $\{a_1, a_2, \ldots, a_m\}$ be an alphabet. A regular expression over $\sum$ is a sequence of symbols formed by repeated application of the following rules:

i. $a_1, a_2, \ldots, a_m$ are all regular expressions, representing the regular sets $\{a_1\}, \{a_2\}, \ldots\ldots,$ $\{a_m\}$, respectively.
ii. $\emptyset$ is a regular expression representing the regular set {}.
iii. ϵ is a regular expression representing **{}**.
iv. If R_1 and R_2 are regular expressions corresponding to the sets L_1 and L_2 then $(R_1 \cdot R_2)$ is a regular expression representing the set $L_1 \cdot L_2$.
v. If R_1 and R_2 are regular expressions corresponding to the sets L_1 and L_2 then $(R_1 \cup R_2)$ is a regular expression representing the set $L_1 \cup L_2$.
vi. If R_1 is regular expressions corresponding to the sets L_1 then $(R_1)^*$ is a regular expression representing the set $(L_1)^*$.

The grammars are not only used to represent a language but still establishing recognizers for them which are basically understood as a convention to define a set. Formed by three parts, an input tape, a finite state control and an auxiliary memory whose type determines its name. The one having a pushdown list would be termed as a pushdown recognizer, more commonly known as a pushdown automaton. In the case of null infinite memory, we have a finite automaton leaving a finite control working along with an input tape only. A nondeterministic recognizer whose maximum memory size is limited to the input size is named a linear bounded automaton.

The formal definition of a Pushdown Automaton, is introduced in terms of Aho and Ullman [V.Aho and Ullman., 1972][1].

Definition 3.1.5. A Pushdown Automaton (PDA for short) is a 7-tuple

$$P = (Q, \sum, \Gamma, \delta, q_0, Z_0, F)$$

where

(1) Q is a finite set of symbols representing the possible states of the finite state control,
(2) $\sum$ is a finite input alphabet,
(3) Γ is a finite alphabet of pushdown list symbols,

[1]Section 2.5, pg.167

(4) δ is a mapping from $Q \times (\Sigma \cup \{\varrho\}) \times \Gamma$ to the finite subset of $Q \times \Gamma^*$
(5) $q_0 \in Q$ is the initial state of the finite control,
(6) $Z_0 \in \Gamma$ is the symbol that appears initially on the pushdown list (the start symbol), and
(7) $F \subseteq Q$ is the set of final states.

Specifying the equivalences [V.Aho and Ullman., 1972] in terms of Chomsky languages between grammars and automata[2]:

(1) A language L is right-linear if and only if L is defined by a (one-way deterministic) finite automaton.
(2) A language L is context-free if and only if L is defined by a (one-way nondeterministic) pushdown automaton.
(3) A language L is context-sensitive if and only if L is defined by a (two-way nondeterministic) linear bounded automaton.
(4) A language L is recursively enumerable if and only if L is defined by a Turing machine.

We close this segment with above correspondences to make clear the range of equivalences that can be handled. Until here all the notions have been collected as a prerequisite to have a comprehensive picture of what will be referred to in some sections of the subsequent chapters. To have a detailed extension of the introduced concepts and theorems there is a vast bibliography, citing something in addition to those already referred to [Hein, 1996], [Linz, 2012], [Sipser, 1997].

3.2 Triangulating a surface

3.2.1 Dividing a field in triangles - Euler problem

The reason is beside our subject here but the formulated problem was, in how many ways a farmland might be divided into triangles. There have been different approaches to solve it. You can find a solution with a clear model, assuming the farmland as a convex n-gon, solved in [Vun and Belcher., 1998] under the title Euler's problem of polygon division.

Being E_n the total number of possibilities in which it can be divided,

$$E_n = E_2 \times E_{n-1} + E_3 \times E_{n-2} + \cdots + E_r \times E_{n-r+1} \cdots + E_{n-2} \times E_3 + E_{n-1} \times E_2$$

which easily gives

$$\sum_{t=2}^{n-1} E_t \times E_{n+1-t}$$

the result is no other than Catalan Numbers.

3.2.2 A triangulated Jordan polygon solved with grammars

As a prior note, here it is convenient to mention that Polish notation refers to a prefix notation in the execution of an operation. We are familiar with the infix notation that we use in arithmetic; a prefix notation means that the order in the operation is: the operator then the operands. Furthermore, by handling binary operations, the grammar results in an unambiguous one.

An elegant and natural way of solving the problem of dividing a polygon into triangles involves the definition of a grammar [Gross and Lentin., 1967], which was ideated by Gross and Lentin whose book was originally published, in French, with the title Notions sur les Grammaires formelles in 1967.

To have it placed, they defined an unambiguous Polish grammar,

$$S \to aSS,\ S \to b \tag{3.1}$$

[2]Plural of automaton.

The symbols S and aSS are related to a directed topological segment and a directed topological triangle respectively. The latter consists of a real side a and two virtual sides S. When we use the production $S \rightarrow\ aSS$ the virtual side S, which is outside the polygon we are composing, has to be replaced by the real side a of the new triangle, paying attention to holding the orientation. Finally, the production $S \rightarrow b$ consists of replacing the virtual side S by the real side b. This is a terminal production which means that the real side b closes the polygon worked. In figures 3.1 and 3.2 we find pairs of parse trees and sectioned polygon. The trees correspond to the words resulted from the building of the polygons beside them. Roman numbers indicate to us step by step how the polygon evolves.

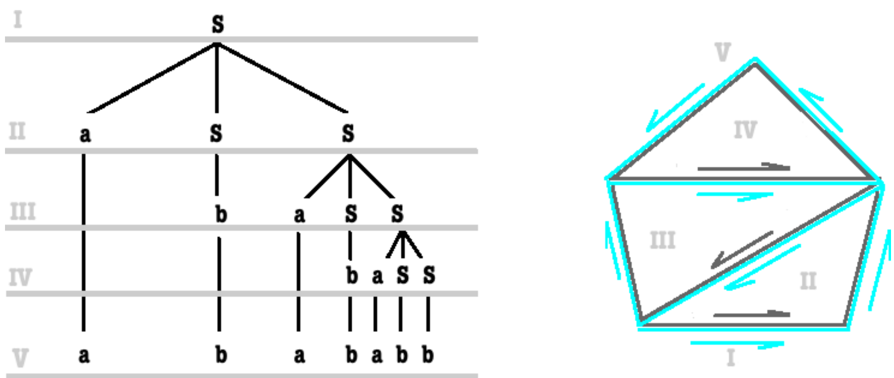

Fig. 3.1: abababb

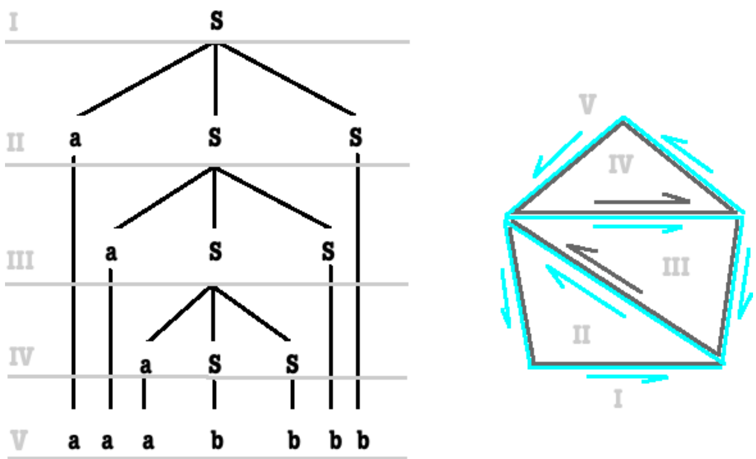

Fig. 3.2: aaabbbb

Let us go bit by bit in the process to get the first derivation illustrated in 3.1. The first step of the method, still cited, is to start with the initial state, in our case S. To this initial state a virtual side is associated along with an addressing which we mark in figure 3.3.A with a cyan color line along with an arrow. The latter will indicate to us what the triangulated section will evolve into. This step is denoted by I [3] in the parse tree as in the triangulated section.

Step two is the first derivation, in the parse tree we use the production

$S\ aSS$. To start with our triangulated section, we replace the virtual side with a triangle formed of one real side and two more virtual ones. Obeying the orientation, couple the new real side with the exact place where before the virtual side, corresponding to the initial state, was. The real side a is associated with a gray line (fig 3.3.B).

In the step number three the contiguous virtual side S will be replaced by a real side b obeying the orientation established to our first real side a, i.e. we are using the production $S \rightarrow b$, which once more is associated with a gray line, and on the resting virtual side the production $S \rightarrow aSS$ will be applied . This is identified in fig 3.3.C. Step IV (fig 3.3.D) consists of repeating step III as many times as it is needed. Last step (fig 3.3.E) finishes the generation of the triangulated polygon replacing the remaining virtual sides on the polygon border by real sides b.

Given an n-gon, the language generated by the grammar defined above to get that polygon will represent the whole set of possibilities to triangulate it. These words will solve the non-commutative equation[4] $S = b + aSS$. The coefficient of the term x^n in the commutative series deduced from it

$$y = 1 + x + 2x^2 + 5x^3 + \ldots$$

will give us the total number of triangles in a polygon of $n + 2$ sides.

Since there is no restriction on the value of n, applying exactly the process explained to

n recursively, any possible triangulated polygon is attainable. The bunch of possibilities of triangulation corresponding to a given polygon is a subspace of the universal flexagon, or ultimate flexagon [Conrad and Hartline, 1962] as it is called by Conrad and Hartline, and each one of these options represents a map of a flexagon.

3.3 Catalan Numbers as the solution of triangulation

From earlier, it had been questioned how many flexagons we can have. Once the polygon system was set, keeping in mind that the polygon system is a generalization of the triangle system which is representing a flexagon map. Furthermore, in turn, it can be understood as the surface's triangulation problem, so the question becomes how many flexagons can be derived given a polygon. Inherently we have a flexagon map for each possible triangulation, then counting the bunch of potential triangulations will give us how many flexagons maps we can obtain.

The issue had been analyzed by Oakley and Wisner [Oakley and Wisner., 1957], as well as by Conrad and Hartline [Conrad and Hartline, 1962]. From the perspective they dealt with it considers discarding some partitions due to these were judged as repeated maps under rotation and reflection operations. Catalan Numbers count the entire possibilities to triangulate a given (convex) area, for what they analyzed and proposed how to discard repetitions from that count, based on their criteria, to have the total of different flexagons that can be obtained.

The tool of the approach presented here to solve that is a grammar, for what the next is noted. Subjected to the restrictions [Gross and Lentin., 1967] imposed by Lentin and Gross, the constructed triangulated polygon in 3.2.2 fig. 3.3 perfectly fulfills the conditions dictated on a triangulated surface to be considered as a map of a flexagon, i.e. every vertex of each triangulated region has to be a vertex of the entire polygon.

Remembering that in the process of obtaining a flexagon pattern, the use of tags is convenient and the fact that they are associated with the vertices and edges of the given flexagon map. Once the tags are assigned, the relationships tag - side and tag - vertex are held until all the

[3] Roman letter.
[4] Detail in chapter 11 of [Gross and Lentin., 1967].

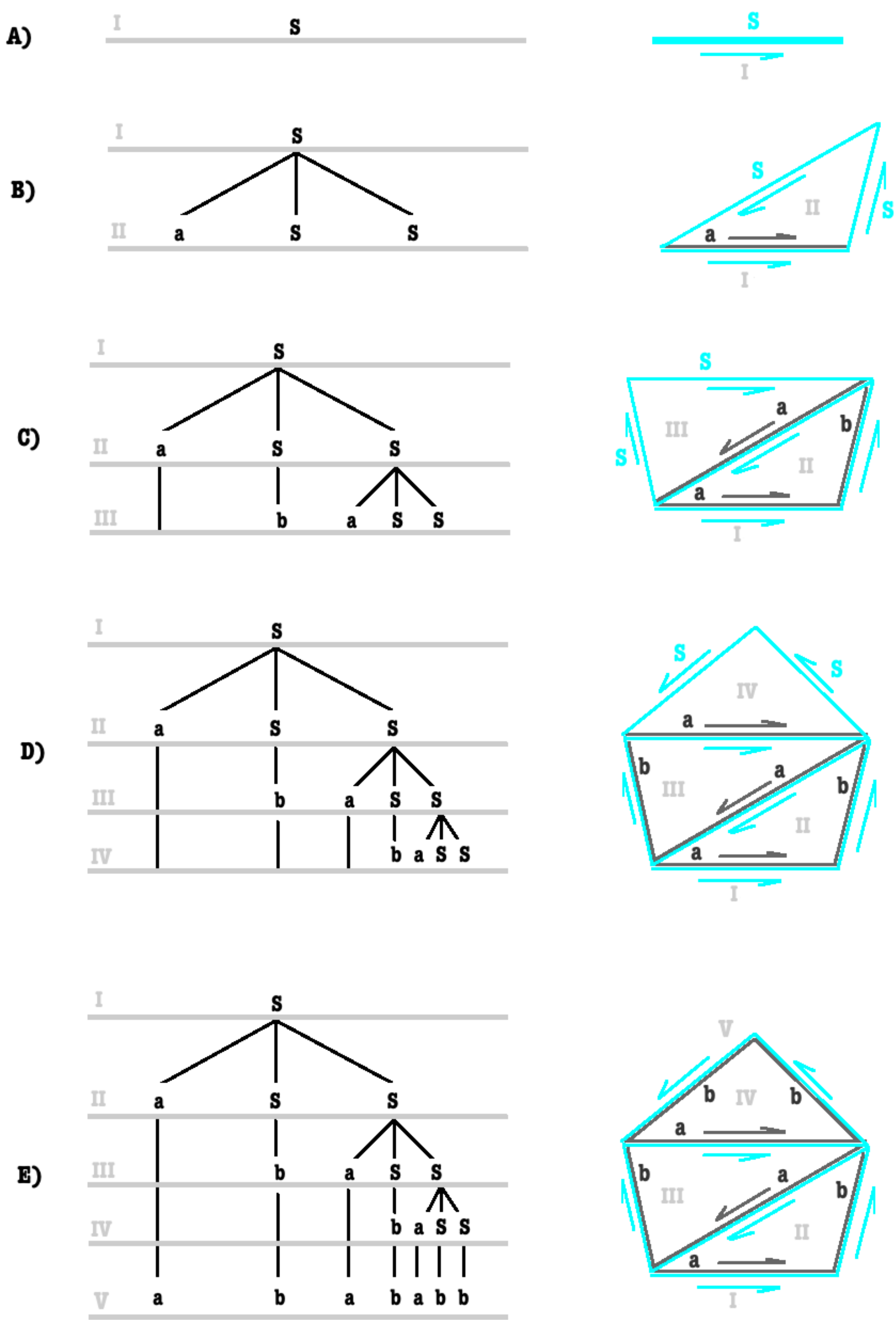

Fig. 3.3: abababb - step-by-step

steps to get the searched partition are completed. Here the zero-position is referring to the side between the first and the last tags, usually numbers. Then we correspond (fig. 3.4) the zero-position to the virtual side corresponding to the starting state S defined in the unambiguous Polish grammar 3.2.2 (3.1) by Lentin and Gross.

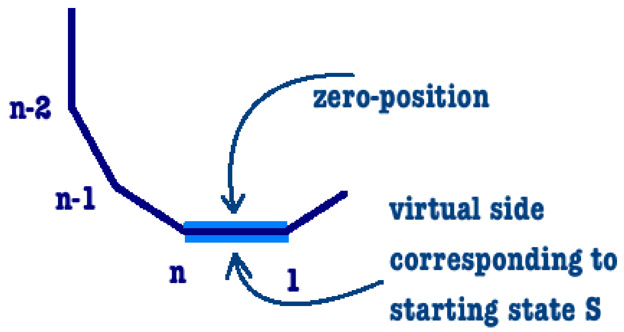

Fig. 3.4: Connection: zero-position - Starting-virtual-side

By rotation or reflection operations, first and second partitions in figure 3.5 could be characterized as the same. However, rotating the second partition conveniently to have the accommodation in the first one but holding the ascribed tags, the virtual side belonging to the starting state S remains the same. Complying with the addressing and the productions dictated by the unambiguous Polish grammar stated in 3.2.2 (3.1) to generate the same second rotated partition, the parse tree will produce exactly the same word as before corresponding to the second non-rotated partition which is different from that one generated by the parse tree of the first triangulated regions' set, the one we tried to get by rotation or reflection.

This scenario shows that that pair of partitions are different when the structure formed by vertices, sides and, tags is transformed as a whole i.e., the triangulated polygon affected by rotation or reflection is subject to the initial tags, and rules to generate it using the unambiguous Polish grammar defined by 3.2.2 (3.1). While the addressing of the readjusted initial state dictates the beginning. This could seem redundant to point the latter, however, it is remarked due to the relevancy to note that it is the root of the corresponding parse tree.

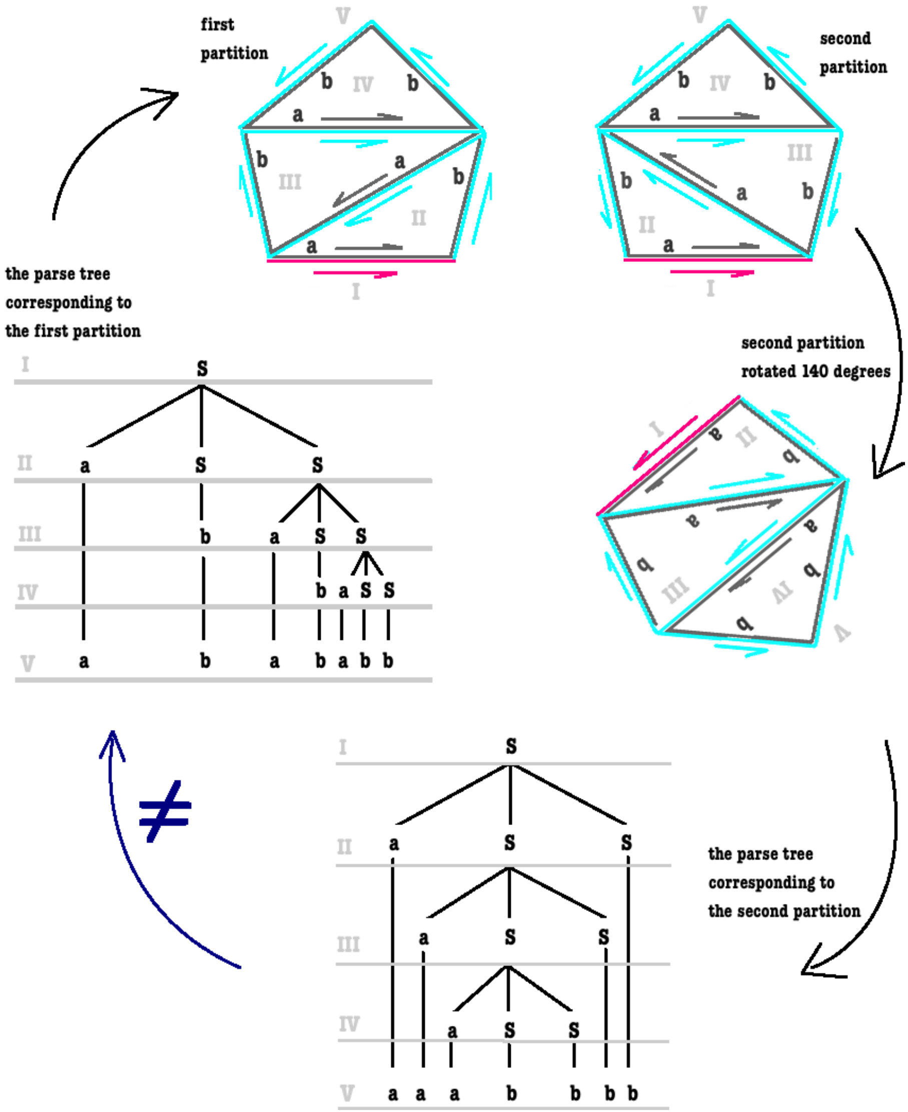

Fig. 3.5: A pair of partitions

Chapter 4

Frieze Code

What does a frieze code is along with how it is obtained are among the central concerns in this chapter. The essentials to understand it are included although you can find a wider explanation into the RIAS publication, "FLEXAGONS" by Conrad & Hartline [Conrad and Hartline, 1962]. In accordance with the material referred to, the term "frieze code" was coined by Harold V. McIntosh, the topic is fully described in [Conrad and Hartline, 1962], however, it was not named like that there. The nature of a frieze code is to be an abstract representation of a flexagon. This is composed of three arrays, two of numbers along with one of signs, these are materialized by a specified method. The trio is disposed horizontally and the sign sequence can be placed on the first or on the third line.

To be more accurate, a frieze code contains the encoded instructions itself to manufacture the pattern to assemble a flexagon. Usually, the upper array of numbers of the frieze code will give us the labels of the front of the frieze, also termed as the pattern, layout or plan and the arrangement of numbers underneath will correspond to the back of it. Basically, the sign sequence guides us where to draw (left or right) the next base polygon, i.e., it regulates what side will be the next hinge.

Because the flexagon extension showed here is by means of maps, a number of complementary notes of the previous efforts on flexagon extension using a type of abstract representation were collected. The vision of Wheeler using circles in the generation of the maps can be understood as a generalization of the use of *n*-gons as if he had predicted that any elementary polygon can be used.

4.1 Frieze Code

Because a frieze code is obtained based on a map, it is convenient to extend its conception. Any polygon subdivided into triangles in such a way that the total of vertices of the triangles remains on the border of the polygon meets the requirements to be considered a flexagon map. The colorful polygon in fig. 4.1, which looks like a vandalized picture, fulfills the condition then it can be regarded to as a map. The polygon was inspired by another one representing a domain homophones to another, all of this in a context of studying the sound produced by drums attending to their shapes[Buser et al., 1994].

Now, let us pay attention to the next polygon in figure 4.2. Each triangulated region was colored separately to note the correspondence between this and the polygon in figure 4.1. The triangulated region in figure 4.2 corresponds to that in picture 4.1 under minimum adjustments which modify neither the order of the triangles nor the correspondence among the angles of the (green) line internally traversing the polygon. Bringing the triangulated area to its graph representation gives us the flexibility to proceed in this way.

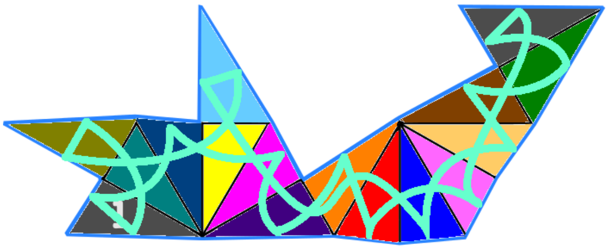

Fig. 4.1: Partial-drum

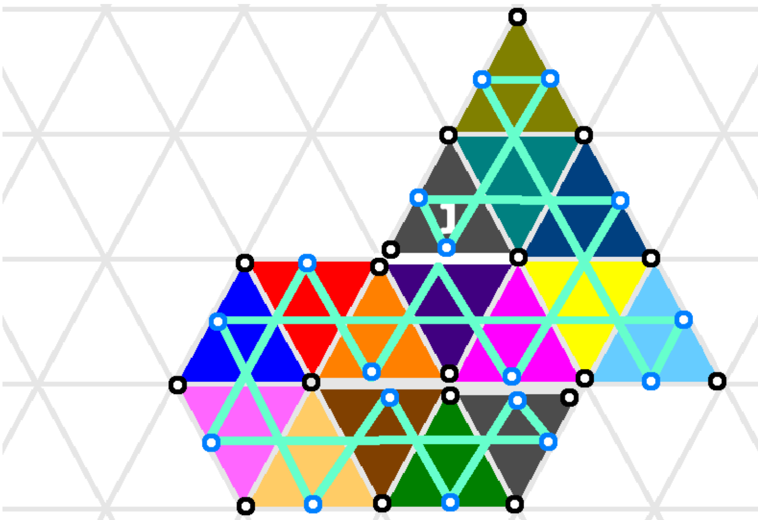

Fig. 4.2: Mapped-to-standard-triangle-net

We should better tidy up the picture and add the data we need to be clearer in the following explanation. First, we unify the triangulated regions forming the polygon to only one color then, we establish the orientation. In this case, counterclockwise is set up as the positive sense. Now we proceed with the basic labeling assigning a number to each vertex of the triangles. We finish ascribing the same numbers but now to the sides exactly as illustration is exemplifying in 4.3, i.e. attaching the minor value to each side between the two vertices.

As we still noted, the frieze code is an horizontal array formed by two rows of numbers along with one of signs. The explanation of how to get those arrangements of numbers is coming up.

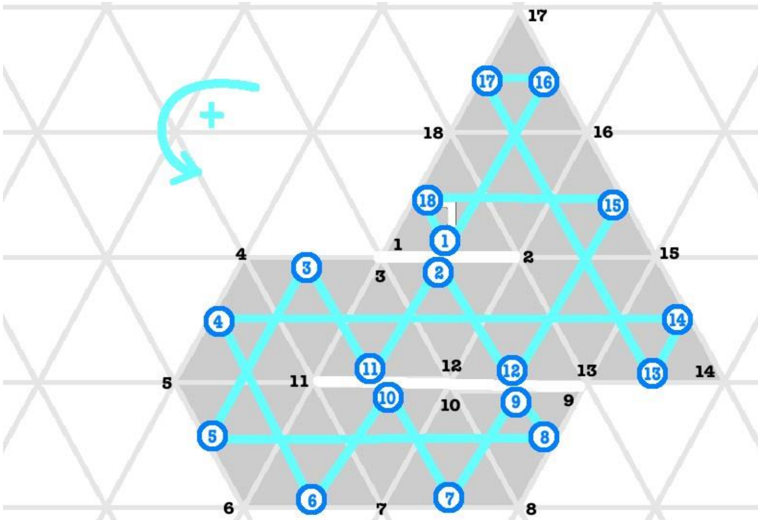

Fig. 4.3: Mapped-to-standard-triangle-net-senseNumbered

The first simple step consists of disposing the sequence of numbers in a row collected along the line according which we traverse it. Base on a horizontal line, the numbers will be deposited alternatively up and down as is showed in fig 4.4-A. Right away we fill empty places with the immediate next consecutive number to that one in the same column. The empty place whose pair corresponds to the maximum value will be the only exception, in this case we will put the minimum value of the complete basic numbering(fig 4.4-B).

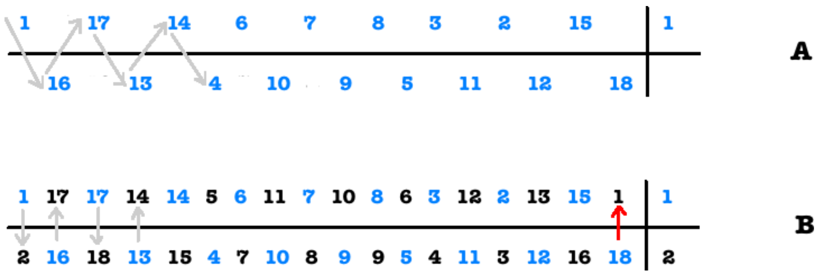

Fig. 4.4: TTDE-CF

To elucidate the final step to complete the frieze code, consider solely the internal line going through the whole polygon to then adding missing orientation. Let us focus our attention on fig. 4.5. A sign was assigned to each vertex according to the positive sense established early in this process. We collect the signs in the same way (order) we collect the numbers, i.e. in accordance with which we find them traversing the line. The figure 4.6 presents the result.

An in-depth study reveals later that a sign sequence could be summing up the suppression of a set of consecutive sides. This transformation is restricted to apply only on the adjacent sides on the border of the map. Nevertheless, this kind of transformation will not be considered in the present work.

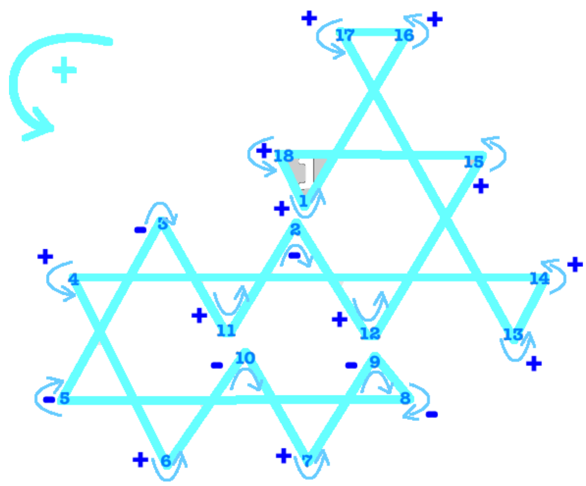

Fig. 4.5: TTD-TTO

+	+	+	+	+	+	+	-	+	-	-	-	-	+	-	+	+	+	+
1	17	17	14	14	5	6	11	7	10	8	6	3	12	2	13	15	1	1
2	16	18	13	15	4	7	10	8	9	9	5	4	11	3	12	16	18	2

Fig. 4.6: TTD-CFs

4.2 Preliminary works on maps to extend flexagons - complementary notes

Continuing with the idea that extending a flexagon means adding faces to it, to attain that there are different ways, summaries of previous efforts to achieve that were still included in section 1.1.3. Once we were familiarized with the concepts related with the frieze code, complementary notes were added here. Once more, focused on the use of abstract representations.

Let us start with annotations about what Wheeler introduced in his article. There, he specifies that he is reproducing in it the explanation contained in the original article of Miss Margaret Joseph appearing in "The Mathematics Teacher for April 1951" [Joseph., 1951]. In short terms, this work describes the generations of hexaflexagrams which in modern terms refers to the pattern to assemble a hexa-hexa-flexagon. Wheeler developed a conceptualization based on it using circles and labels denoting colors and the derived ones according to their combination with each extension, this to paint and assemble a flexagon. We have still illustrated some figures (M1 1.6, M4 1.7) in appearing in his work, "The Flexagon Family" [Wheeler., 1958].

Starting with primary colors, the pictures M1 1.6 and M4 1.7 show the enlargement of flexagons not only by maps but by recursion, from a tri-flexagon to having a hexa-flexagon with the secondary colors and then obtaining a hexahexa-flexagon with the tertiary ones. To present the next level he changed notation from letter (associated with colors) to numbers. Additionally to introduce the Miss Margaret Joseph's work, furthermore, Wheeler tried flexagon's extension different from a uniform cyclic growing from the flexagon map as they are shown in figures 4.7 and 4.8.

Let us move directly onto the revision of the results of Conrad and Hartline and how the frieze code is affected with joining. The picture 4.9, originally borrowed from [Conrad and Hartline, 1962] with Tukey Triangle Network, added to identify easier how the transformation in the map modifies its unit plan. A pair of maps and their corresponding couple of arrangements to label the unit plans are shown. To note, the elementary triangle is added to the original zero-side, the splitting on the map is reflected in the array of numbers as a column splitting and rearranging the result as corresponds, this is remarked in the number collection associated with each unit plan.

Although, they particularly revised the illustrative example of coloring[1] scheme, to give us a better idea of how the full process of splitting, removing and replacing is working on a straight strip flexagon, the authors predicted that this same process might be utilized in whatever sort of flexagon.

Conrad and Hartline utilized [Conrad and Hartline, 1962] the polygon system as the medium to extend flexagons, remembering that it is the generalization of the Tukey Triangle System. They used it as a shortcut to extract information, but mainly to show evidence the correlation between the sequence number and the map annotating that this system works for proper complete flexagons due to it having only two choices of hinges.

McIntosh directly normalized flexagons maps [McIntosh, 2012] graphs, instead of their corresponding Tukey Triangle Networks and proposed collecting data from the joined maps after normalizing the joining. This procedure is not restricted to straight strip flexagons. In fact, it can be applied to any kind of flexagons maps whose elementary polygons are triangles.

4.3 The McIntosh method, normalizing for joining

Current section bring together flexagon maps in conjunction with a normalization procedure still revised to make clear how they work to have the joining of a pair of flexagons, let us refer to this as the McIntosh method [2]. By this time the fact was still introduced that normalizing procedure

[1] Using numbers as tags.
[2] See Preface, page v

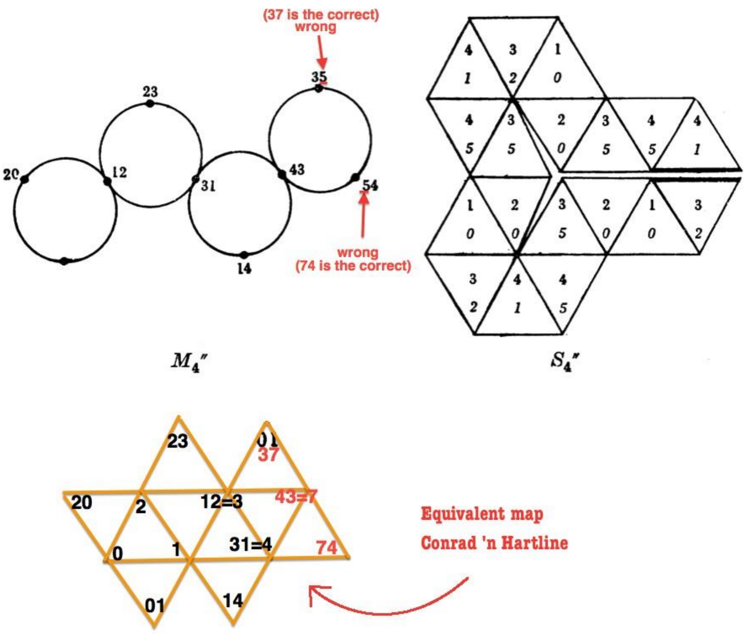

Fig. 4.7: Wheeler(M4") - Conrad 'n Hartline equivalent maps

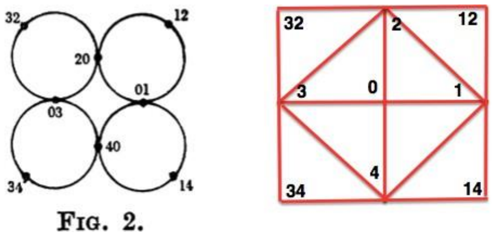

Fig. 4.8: Wheeler(Fig2) - Conrad 'n Hartline equivalent maps

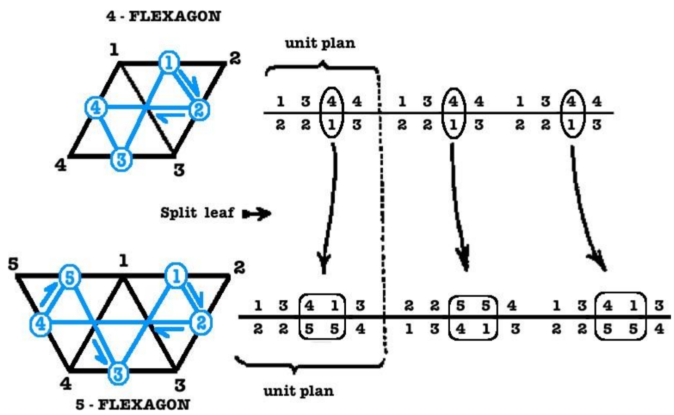

Fig. 4.9: Frieze-Code-affected-by-splitting-attaching

is applied directly on the elements of the graph representing the map of a flexagon.

The idea is quite similar to what was described by Hassler Whitney (section 2.2.5) in "a Theorem on Graphs" [Whitney, 1931] paper. In this case, the vertices are the ones representing a flexagon map. We proceed rearranging them circled around while initial connection among them are holding, provided that we are not assigning any weight value to the edges the fact of shorting or stretching them is not relevant.

Notice that, all the sides staying on the border of the flexagon map graph will now draw a polygon and the sides connecting them internally in the original flexagon map graph will be the triangulated regions referred to in the section 3.2.1 as the Euler Problem as well as in section 3.2.2 solving it, the triangulation problem, via an unambiguous Polish grammar. The picture will give us further detail. And figures A and B of 4.10 and 4.11 exemplify the transformations of two flexagon maps.

Step C of 4.10 and 4.11 is anticipating the next movement. They are the dispositions belonging to the second normalization already distributed in the corresponding half of the joining polygon. They were reorganized, again, to form only one polygon, the one we have after joining the given maps. At this point, we can foresee that the way of collecting the data will draw a kind of "eight or ∞ symbol" inside the joining and that this will happen to each joining. Figure 4.12 shows us what we mean with this expression.

Once we had the second normalization and the joining has been completed, we proceed to generate a second triangulation which will correspond to connect middle points among existing sides, these are the equivalent of the Tukey Triangle Network, and they will determine the way in which the data will be collected. Pictures 4.13 and 4.14 clarify how it is done.

To build the frieze code of the resulting joining, we carry on following the final curve drawn inside the internal second triangulated regions (fig. 4.15) collecting the number sequence and arranging them, again, alternating above and below a line exactly as it was done to the frieze code in the first section of this chapter, the individual ones are indicated too. Now we complete adding "one" to each corresponding position. Finally we collect the signs in the same order as the number sequence was collected. The full resultant example is in figure 4.16, containing:

- **i)** A pair of polygons joined and normalized in the resultant polygon, the connection is illustrated by the gross red line.
- **ii)** Triangulation traced in pink lines.
- **iii)** Cyan lines delineates the dual.
- **iv)** Gross turquoise curve indicates the go through of the first polygon. The second one is traced by a gross green curve. And the orange curve is composing the full traverse. Corresponding colored curved arrows denoting their orientation.
- **v)** Frieze codes, three arrangements resuming flexagon data to the first, second and joining poly- gons.

Here it is convenient to point out the work of Madachy [Madachy, 1979] presented as "structure diagrams", they look quite similar to the model delineated as part of the McIntosh method[3], however, do note that the steps they follow and the concepts they introduced to generate them are distinguishable. Finally, it must be revealed that there were others efforts having subtle differences handling the graphs corresponding to the flexagon maps. Appendix B is exposing pictures illustrating that, those were taking borrowed from the original located in folder corresponding to the class though by McIntosh in the "XXXIV Summer's School" [McIntosh, 2011].

[3]See note in Preface.

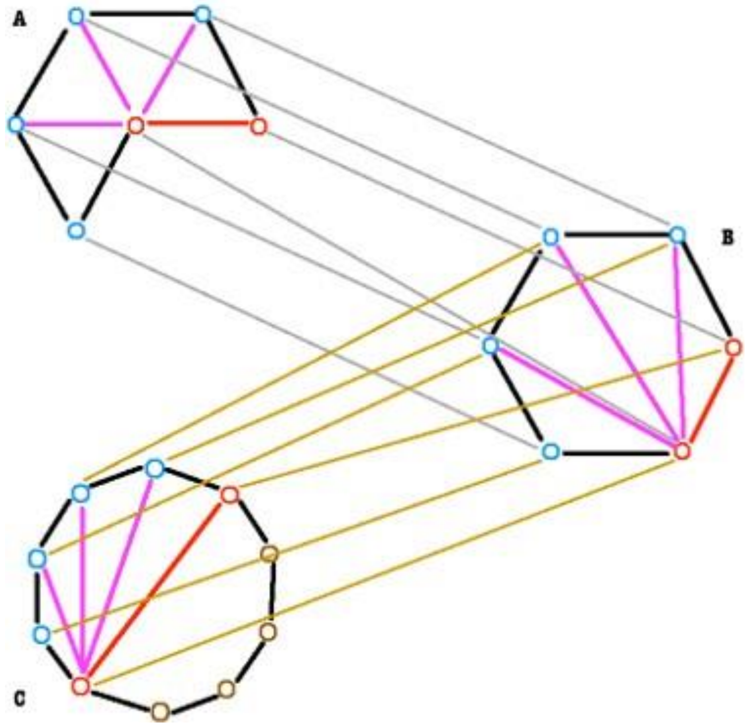

Fig. 4.10: ConradP - normalized

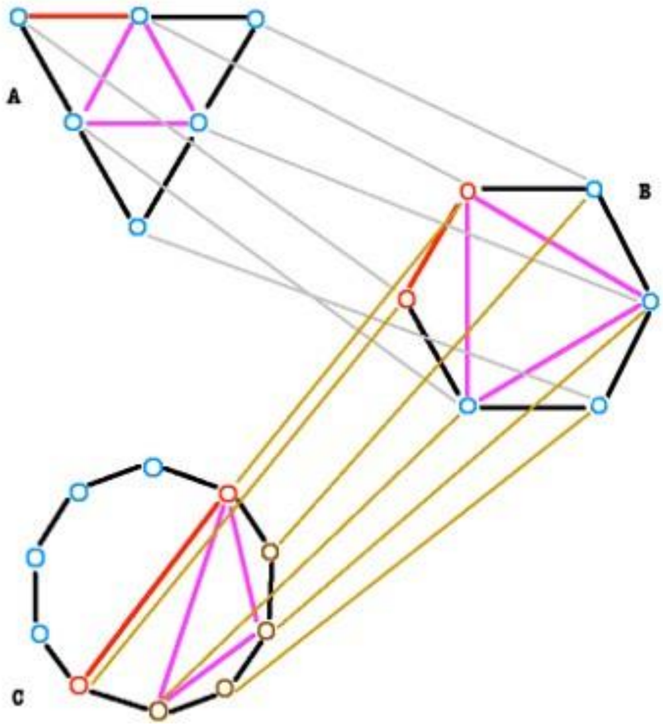

Fig. 4.11: ConradT - normalized

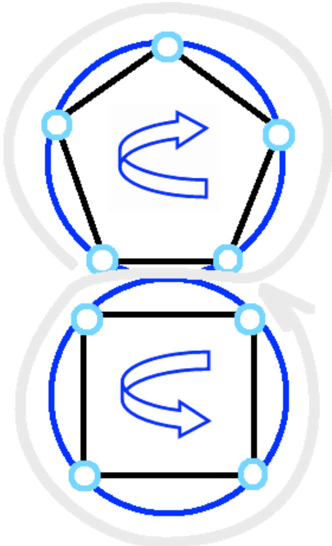

Fig. 4.12: Penta - Tetra path Infty

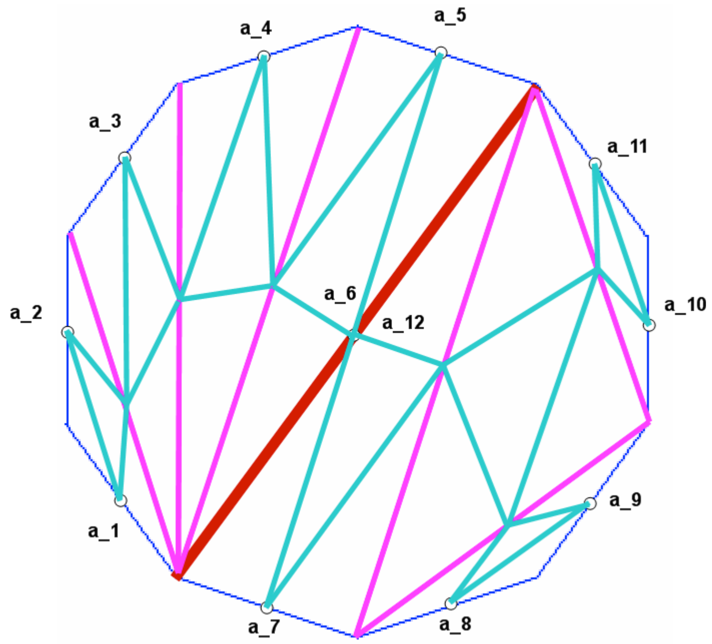

Fig. 4.13: DecaIi Union

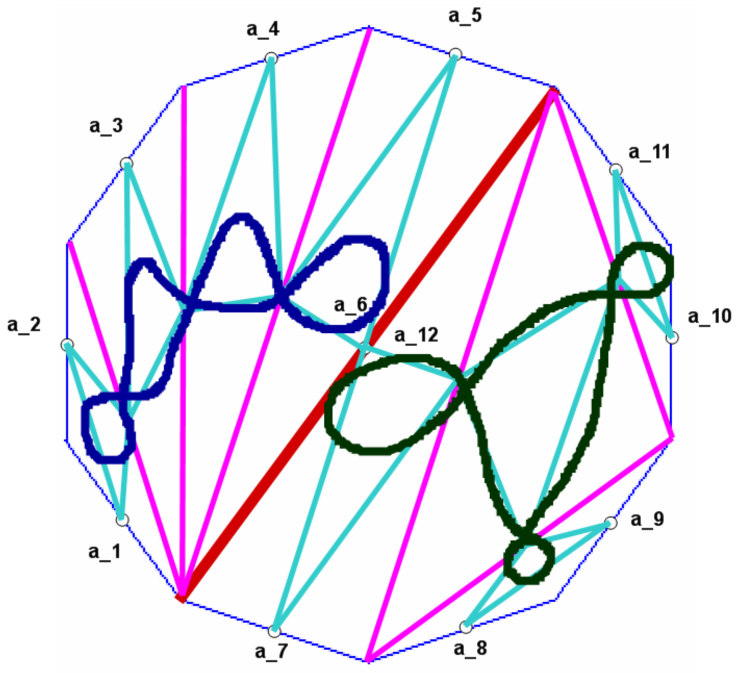

Fig. 4.14: DecaIi3

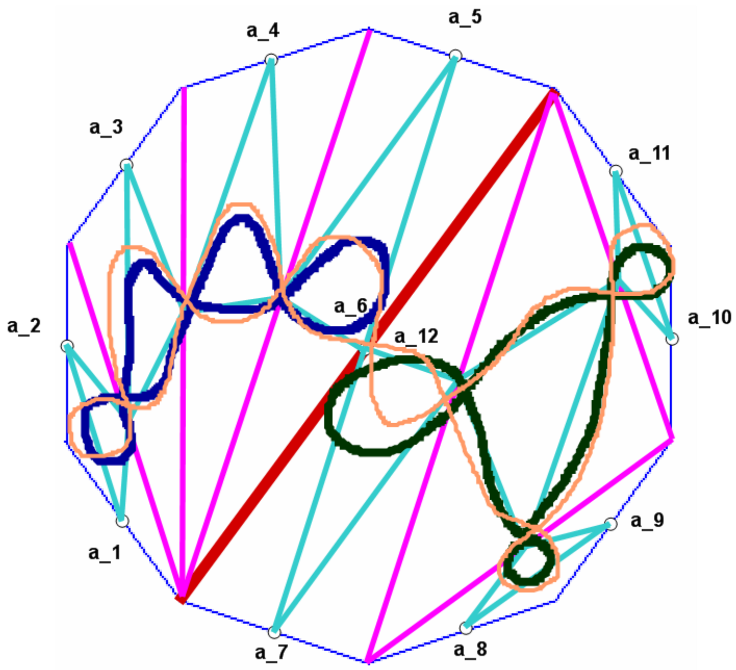

Fig. 4.15: DecaIi4

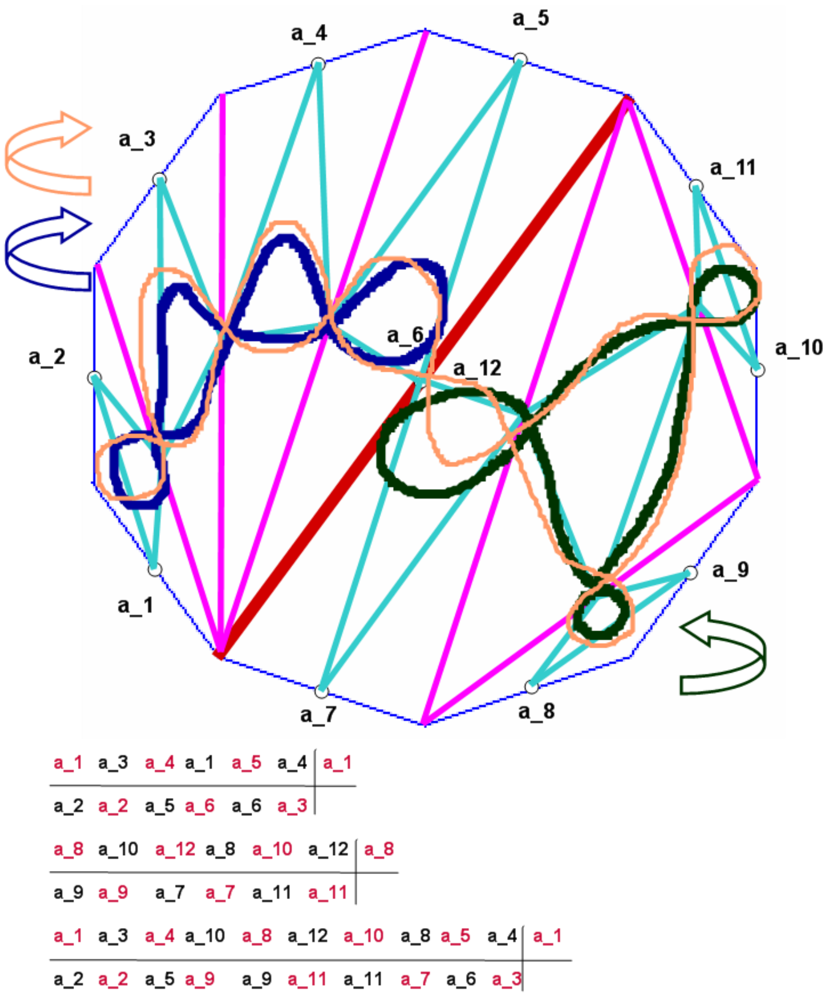

Fig. 4.16: DecaIi4i

Chapter 5

Closure property under substitution

This chapter not only reports the closure property under substitution operation to the unambiguous Polish grammar[3.1] defined to generate the general triangulated polygon [3.2.2], but makes use of it alongside other concepts to have the closure property to flexagons. For such purpose, we will extend those ideas and concepts summed up in section 3.1. Substitution operation is added to the closure property of union and concatenation handled in the elements used to generate languages. The theorem[a] declared, and proofed to the context-free languages, encloses a result which is essential to show the closure property to flexagon (maps), the subject of this work. This operation is a language-valued as [Hopcroft et al., 2001] and [Shallit, 2009] refer.

Current section was developed based on the perspective of Carroll and Long [Carroll and Long, 1989] relating it to the convenient outlook introduced when unambiguous Polish grammar was defined to solve the problem of triangulating [Gross and Lentin., 1967] a given polygon whose result satisfies being a map of a flexagon.

[a]Section 6.2, pg.131

5.1 Substitution Operation in Context-Free Languages

Because later it will be referred to, we immediately introduce the concept of power set.

Definition 5.1.1. Let be A a set, the power set of A designed by 2^A, even extendedly denoted by $\wp(A)$ is

$$\wp(A) = \{X \mid X \subseteq A\}.$$

Treating flexagon maps in terms of grammars, the type of extension we are pursuing on flexagons is matched up to the substitution operation running on context-free languages. The formal definition of the substitution operation introduced right away is the one presented by Carroll and Long [Carroll and Long, 1989][1]:

Definition 5.1.2. Let S = $\{a_1, a_2, ..., a_m\}$ be an alphabet and let G be a second alphabet. Given context-free languages $L_1, L_2, ..., L_m$ over G, define a substitution $s : \Sigma \to \wp(\Gamma^*)$ by $s(a_i) = L_i$ for each $i = 1, 2, ..., m$, which can be extended to $\bar{s} : \Sigma^* \to \wp(\Gamma^*)$ by

$$\bar{s}(\lambda) = \lambda,$$

and

$$(\forall a \in \Sigma)(\forall x \in \Sigma^*)(\bar{s}(a \cdot x) = s(a) \cdot \bar{s}(x)).$$

$\bar{s}$ can be further extended to operate on a language $L \subseteq \Sigma^*$ by defining $\bar{s} : \wp(\Sigma^*) \to \wp(\Gamma^*)$ where

$$\bar{s}(L) = \bigcup_{z \in L} \bar{s}(z).$$

As it was already observed (section 3.2.2), the interpretation of Carroll and Long introducing unambiguous Polish grammar deciphering, in terms of grammars, the question of triangulating a polygonal surface satisfies the condition to be a flexagon map. Such grammar,

$$S \to aSS \mid b$$

under hierarchy's Chomsky leaves into the context-free grammar set. Let us introduce in fig 5.1 the (nondeterministic) pushdown automaton recognizing it. As part of the background we need

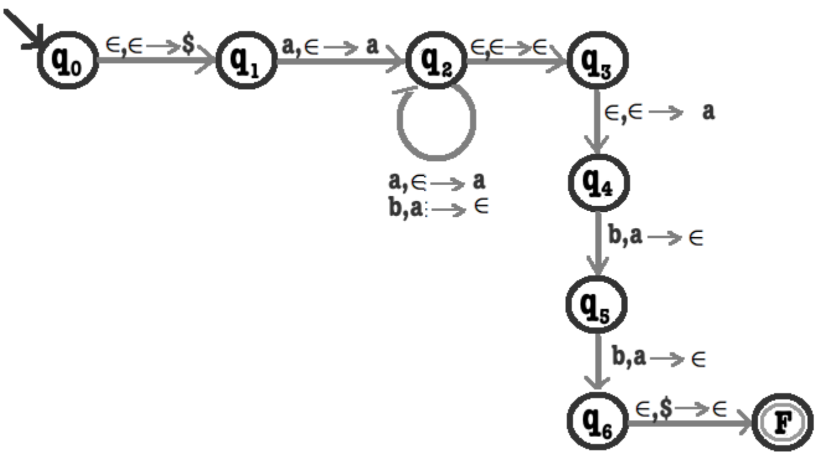

Fig. 5.1: The recognizer (PDA) to the unambiguous Polish Grammar

to show the closure property to flexagons (maps) under substitution operation is the theorem[2], in [John E. Hopcroft, 1979].

[1]Section 9.4, p.319.
[2]Section 6.2, pg.131

Theorem 5.1.1. *The context-free languages are closed under substitution.*

Observe that having a pair of triangulated polygons generated according to the unambiguous Polish grammar defined in 3.1, figure 5.2 is stressing the triangles in turn when linking takes place. In addition, the starting virtual sides and their orientation and then the possible side(s) where the starting side can be shifted after attaching are signaled. Note it is restricted to keeping the same orientation and being a side of one of the triangles directly involved in the joining, staying on the polygon's border after the linking. For the remaining part of the given polygons it is sufficient to know that it is generated following the grammar defined. Caring colors will help us to match the growing of the parse tree with its corresponding polygon. Picture 5.2 sums up all possibilities in which they could be disposed at joining.

To exhibit their behavior during this process let us outline their corresponding parse trees. Note that the branches related to the dashed part of the polygons were not explicitly built but signaled with subtrees. The parse trees after joining are built based on the redefined starting side marked with a colored external arrow. The first two complete samples of derivation are illustrated in figure 5.3, the rest were included in appendix C. Based on them we proceed to identify the rule's substitution.

Let it be Λ_1 and Λ_2 the full derived parse trees corresponding to the first and second triangulated polygons. With λ_1 and λ_2 being the partial parse trees corresponding respectively to the rest of the first and second polygons do not consider the triangles directly involved in the joining. We know that Λ_1 and Λ_2 are in the context-free language corresponding to the unambiguous Polish grammar 3.1, proposed by Lentin and Gross [Gross and Lentin., 1967]. Due to the recursive nature of the defined grammar, the same can produce the referred sub-trees λ_1 and λ_2. Let L_1, L_2, L_3 and, L_4 be all subsets of the same language produced by the unambiguous Polish grammar, generating Λ_1, Λ_2, λ_1 and, λ_2 in same order.

It is claimed that proper flexagons are closed under attaching, in other words, a flexagon is obtained from joining two others. To proof that it is used the theorem 5.1.1. To make this approach clear let us put together the central notions, already introduced, to connect them. The essential prerequisite to consider a triangulated polygon as a flexagon map candidate is that the totality of its vertices stay on its border. Lentin and Gross resolved [Gross and Lentin., 1967] the question of "in how many ways a Jordan [3] polygon with labelled vertices can be partitioned in triangles?" introducing an unambiguous Polish grammar 3.1. The solution was conditioned to have non-intersecting diagonals, observe the resulting triangulated surface ideally satisfies being a flexagon map. Notice that treating the flexagon maps as graphs allows us to stretch or elongate their sides to dispose in a suitable convex polygon. To complete the thought, consider the McIntosh method reached the joining of flexagons by means of their maps, not before putting them in theirs corresponding normal graphs.

Extending flexagons by means of joining their maps defined as by the unambiguous Polish grammar[3.1] and obeying indications to reassign (shifting) the virtual starting side after merging them was identified with the substitution operation on context-free grammars. Let us define one more pair of λ's, λ_3 and λ_4 denoting a and b respectively. Establishing $\Sigma = \{a, b\}$ and $\Sigma' = \{\Lambda_1, \Lambda_2, \lambda_1, \lambda_2, \lambda_3, \lambda_4\}$ the alphabets being used and the substitution rules:

i) $S(\lambda_3) \longrightarrow \lambda_3$, only in first position.
ii) $S(\lambda_4) \longrightarrow \lambda_1$, if zero-side is set on the second polygon.
iii) $S(\lambda_4) \longrightarrow \lambda_2$, if zero-side is set on the first polygon.
iv) $S(\lambda_2) \longrightarrow \Lambda_1$, if zero-side is set on the second polygon.
v) $S(\lambda_1) \longrightarrow \Lambda_2$, if zero-side is set on the first polygon.

Although it was said that all combinations were exposed, in fact only the well-behaved combinations in the arrangement of the joining were included. With well-behaved combination are being referred the ones where a contiguous side of the z-side is available to be considered as the root

[3] [Gross and Lentin., 1967] a simply topological polygon with an inside and an outside.

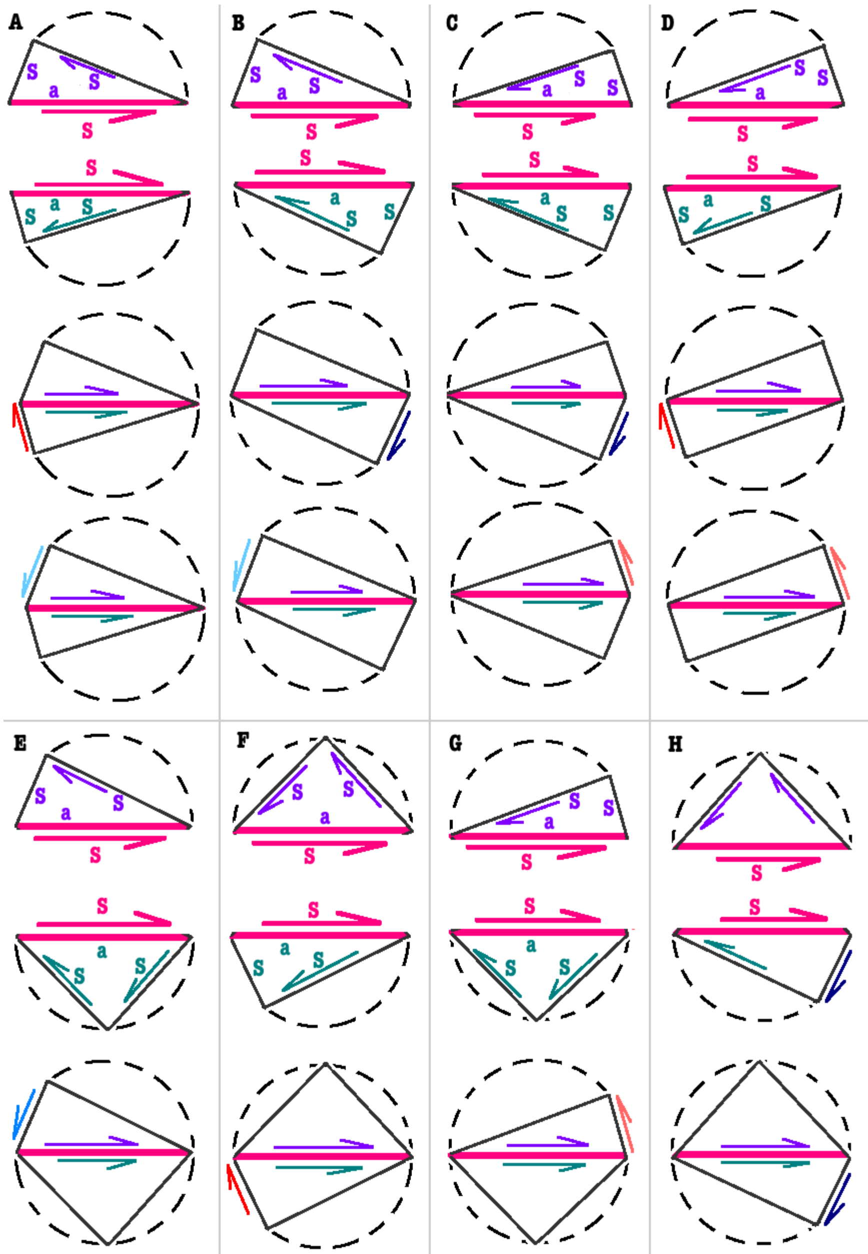

Fig. 5.2: All-combination-disposition-of-triangles-in-joining

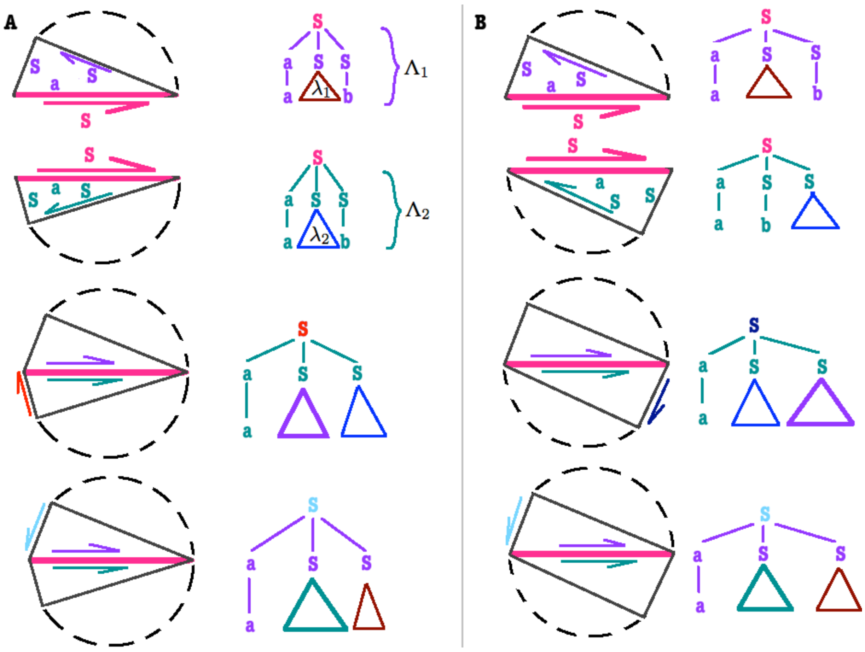

Fig. 5.3: JoinPosTree-i1-2l

after merging is completed. Such case, illustrated in figure 5.4 was excluded above, is approached right now.

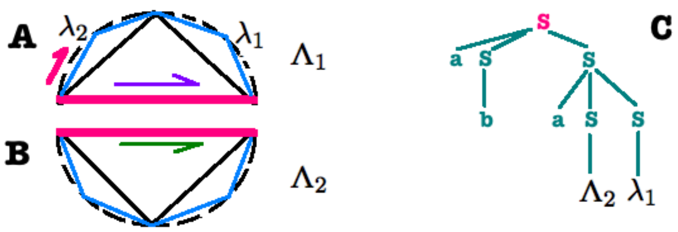

Fig. 5.4: Case-bad-behaved

In addition to the λ_1, λ_2, Λ_1, and Λ_2 presented in 5.4, alternatively denoting *a* and *b* as λ_3 and λ_4 respectively. The new reassigned root is marked with a gross pink arrow, left of the first polygon. The parse tree is included to stress we are going one level deeper to have the substitution rules. The suggested for this situation are:

i) $\lambda_3 \longrightarrow \lambda_3$, only in first position.
ii) $\lambda_1 \longrightarrow \lambda_4$
iii) $\lambda_2 \longrightarrow \lambda_3\Lambda_2\lambda_1$

Once the substitutions rules were declared and the set of the context-free languages exposed, due to theorem 5.1.1 it can be concluded that triangulated flexagon maps are closed under substitution operation. Cases of 1 and 2 sides are completely out of this context, so the minimum case considered here is an elementary polygon with three sides, however, the statement done is wider, therefore the above exposed is the half of what we need to show. To the complementary inductive part, let us come back later in section 5.4. Meanwhile, section 5.3 introduces the grammar to do so but not before section 5.2 tightens what exposed above with one example.

5.2 One example

Right away a step by step example is presented to make clear the way it works. Assuming the alphabets Σ' and Σ as they were defined. Figure 5.5 illustrates:

i) A_1 is the first flexagon map with the starting side signaled.
ii) A_2 is the Jordan polygon generated from the unambiguous Polish grammar 3.1, the normalized representation of the flexagon map.
iii) $\wedge_1$, the entire parse tree of the first polygon.
iv) B_1 is the second flexagon map with the starting side signaled.
v) B_2 is the Jordan polygon generated from the unambiguous Polish grammar 3.1, the normalized representation of the flexagon map.
vi) λ_2, the partial parse tree of the second polygon.
vii) C_1 is the normalized polygon after linking.
viii) C_2 is the parse tree of the polygon after joining, pointing out the $\wedge_1$(green triangle) and λ_2(brown triangle).
ix) D_1, D_2 are the half part for the first and the second normalized polygons. D_3 is the type of attaching.
x) $\wedge_1, \lambda_2, \lambda_3, \lambda_4$ were put where corresponding showing how substitution is done.

Once certain familiarity is achieved, it is possible to skip some steps. Figures D.1, D.2 and, D.3 in Appendix D are illustrating one more example. It is advisable to refer to it to see what is happening directly on the flexagon maps during the joining process under grammar conception.

5.3 Redefining the grammar

Relating exactly as Lentin and Gross did, S to virtual sides and $\{a, b\}$ to real ones and redefining the grammar as

$$S \to aSS^+,\ S \to b \tag{5.1}$$

that is:

$$S \to aSS^+|b$$

This subtle modification allows to generate elementary n-polygons. The bunch of figures in 5.6, 5.7, 5.8 are a sample of how it works. These are only showing uniform flexagons maps, i.e., all elementary polygons are of the same number of sides. However the recent defined grammar allow to incorporate elementary polygons of different number sides in the same flexagon map.

Let us start by outlining the idea that this grammar 5.1 makes possible to generate any flexagon map. That it is conceivable to have mixed elementary polygons as part of one same flexagon map was already declared, the explanation is that once a rule production was applied to each branch having non-terminal symbols will evolve independently from each other with the only restriction of obeying the rules of production defining the grammar in turn.

A flexagon map with uniform elementary polygons is drawn, initially restricting the reassignation of the starting virtual side to the immediately contiguous side, the one determined following the addressing of the preliminary starting virtual side. Figures 5.9, 5.10 reveal how substitution takes place, λ_1 replaces the first λ_4 from left to right, the one immediate to the first λ_3 and the last λ_4 will be removed to make room to $\wedge_2$.

The previously described behavior will be characteristic of the case of a quite-chain which involves in substitution the first and last branches λ_4 in the parse tree. Removing the restriction on the redefinition of the new virtual starting side of being the immediate one next to the original following its orientation. For the general case, the orientation of the original starting virtual side must still be held and the replacement rules must be conveniently dictated according to the redefined virtual side.

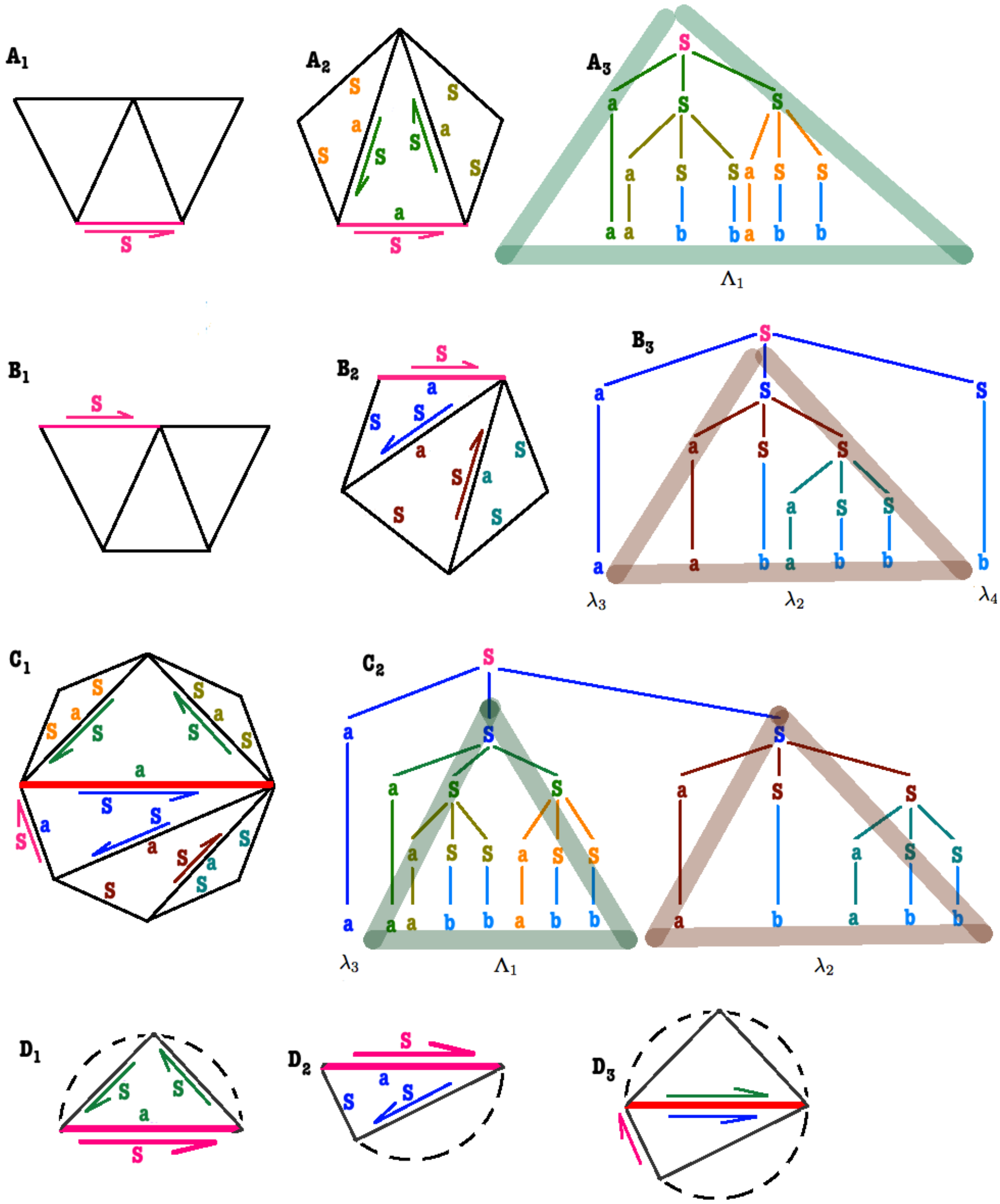

Fig. 5.5: Ex13-17

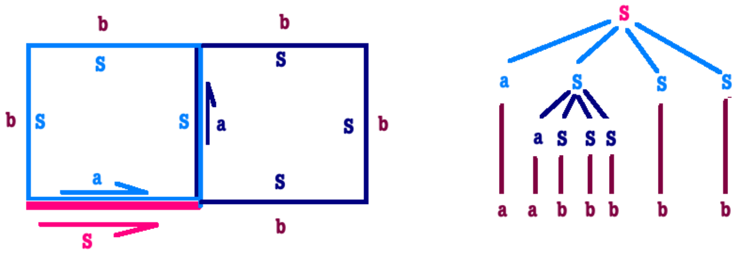

Fig. 5.6: Tetra-grammar

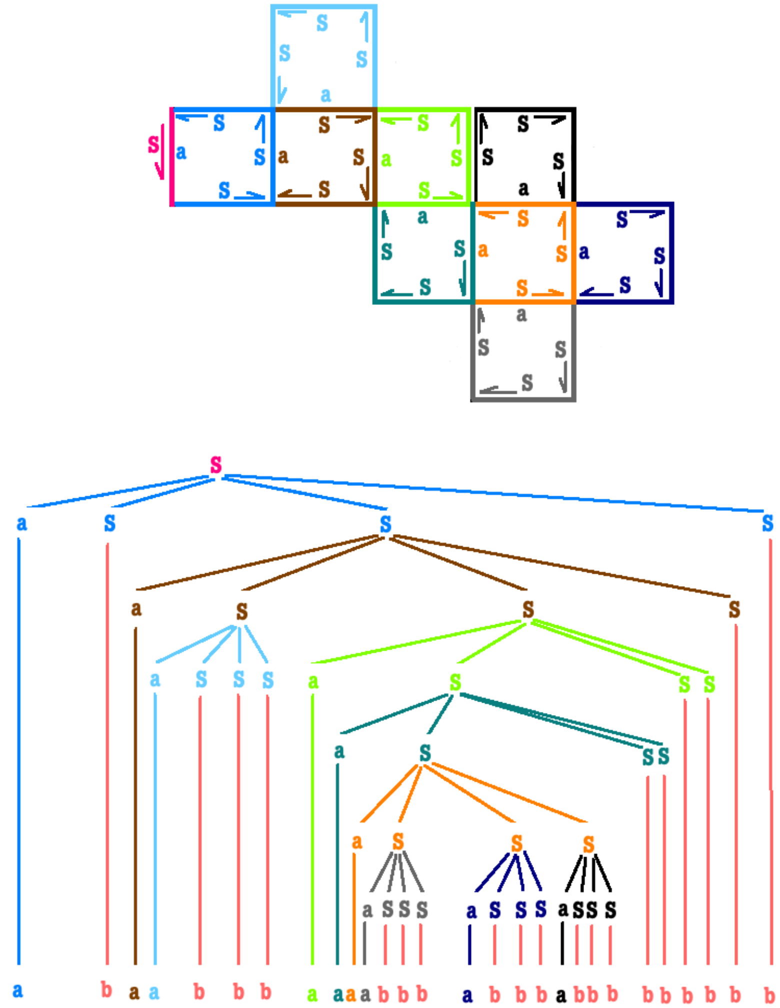

Fig. 5.7: Tetra-grammar-c

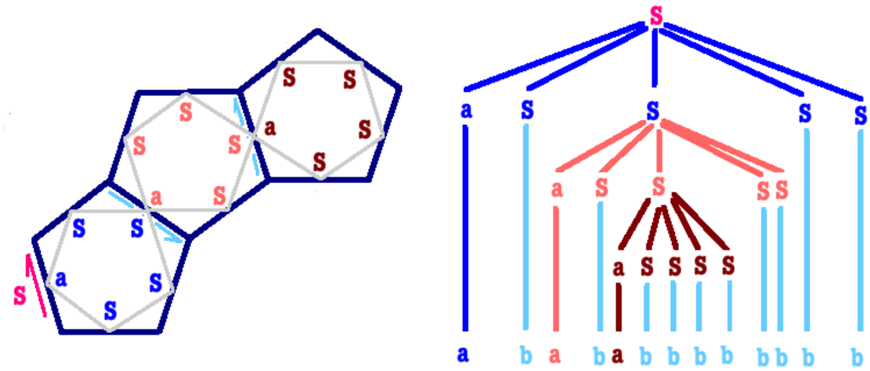

Fig. 5.8: Penta-grammar

5.4 Complementing Substitution Operation in Context-Free Languages

Once the unambiguous Polish grammar[5.1] thanks to Carroll and Long [Carroll and Long, 1989] was redefined 5.3, to do possible the derivations of flexagon maps constituted of n-polygons where $n > 3$,

$$S \to aSS^{+}|b$$

let us move on to the (nondeterministic) pushdown automaton recognizing the redefined grammar which is illustrated in picture 5.11. Now, for the inductive part, let us suppose true the case for n-polygons and show it is hold to the case of $n+1$-polygons. The well-behaved combinations to join are those where at least one of the polygons has a side adjacent to its z-side leaving in the frontier of the resulting polygon after joining. Due to it is a good candidate to be the root of the recently obtained polygon. When we have n-gons the patterns of the parse trees are like that in 5.12.A and 5.12.B, the merging can be AA, AB, BA, BB according to the disposal of the polygons prepared to be joined. Once more, renaming a as λ_3 and b as λ_4, being constituted λ_1, λ_2, $\wedge_1$ and $\wedge_2$ as those exhibited in 5.12.A and 5.12.B. To the case of n-gons and $n+1$-gons, the rules of substitution can be simplified to:

i) $s(\lambda_3) \to \lambda_3$, only to the first position.
ii) $s(\lambda_4) \to \lambda_2$
iii) $s(\lambda_2) \to \wedge_1$

Only rest to list the substitution rules corresponding to the case of no-well-behaved. Observe fig 5.13.A and 5.13.B show the set of λ's from 1 to n - 1 correlated with the partial sub-trees and the absolute $\wedge$'s tagging each polygon 1 and 2 disposed to be linked. As we have n -1 precedent λ's, a and b will be designated as λ_n and λ_{n+1} respectively. Readily, the set of substitution rules related to excluded case corresponding to n-gons, from which immediately can be dictated the ones suggested to the case $n+1$-gons. Bear in mind that the recursive nature of the grammar makes it possible to generate with this same each sub-tree tagged with the λ's, that is, to generate the derived languages.

i) $s(\lambda_n) \to \lambda_n$, only to the first position.
ii) $s(\lambda_1) \to \lambda_{n+1}$
iii) :
iv) $s(\lambda_{n-2}) \to \lambda_{n+1}$
v) $s(\lambda_{n-1}) \to \lambda_3\ \wedge_2\ \lambda_1\ \lambda_2 \ldots \lambda_{n-1}$

Those sets of substitutions rules complementing the general case along with the set of the context-free languages exposed, and then due to theorem 5.1.1, it is concluded that n-gons flexagon maps are closed under substitution operation. Now the idea of having a covering space from which any flexagon can be obtained is clearer.

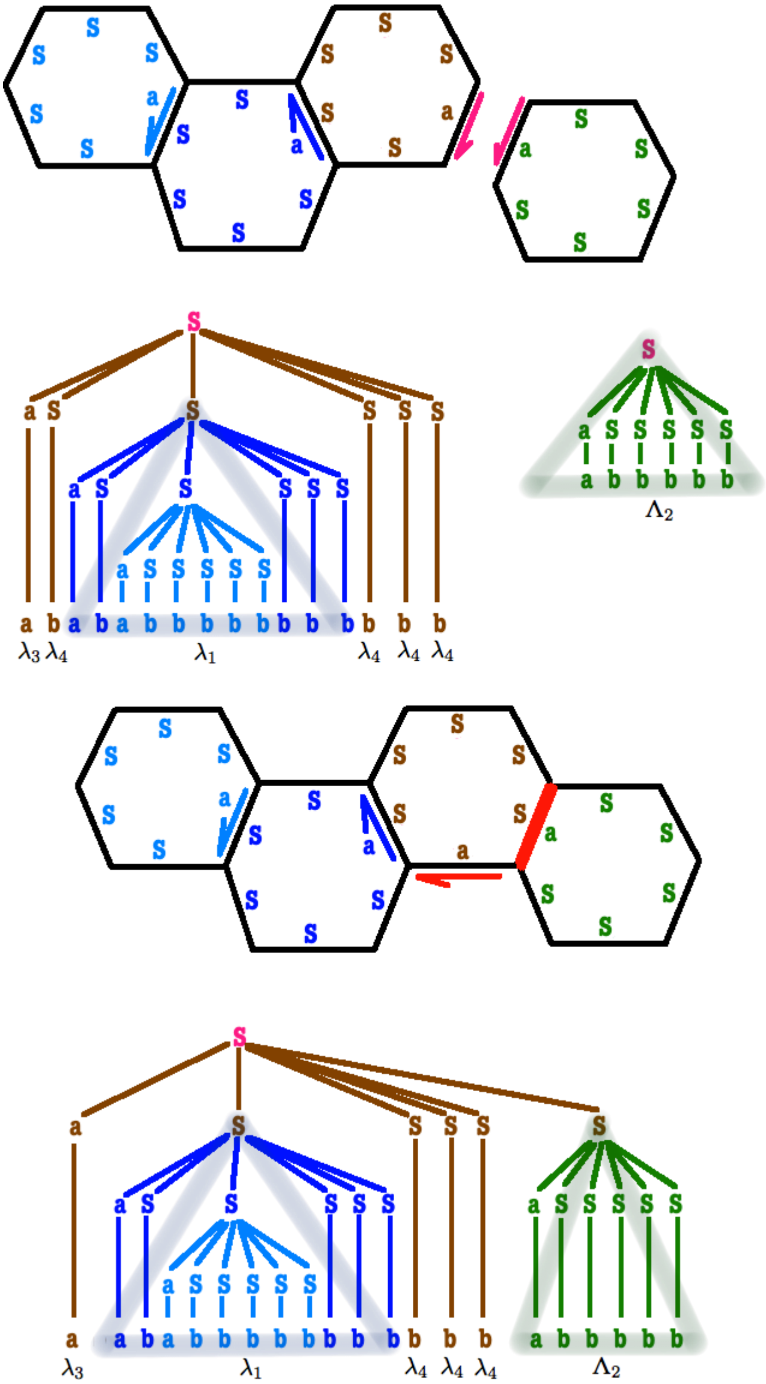

Fig. 5.9: Hexa-mapAttach-PT

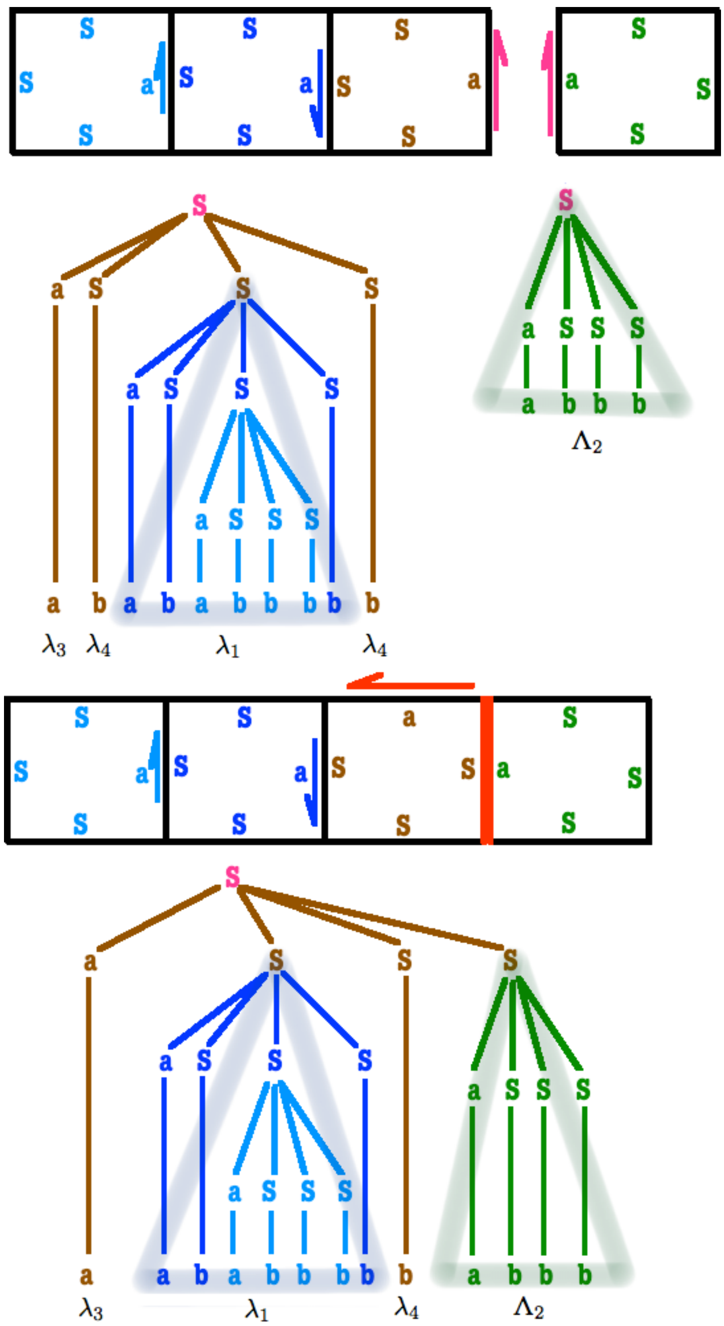

Fig. 5.10: Tetra-mapAttach-PT

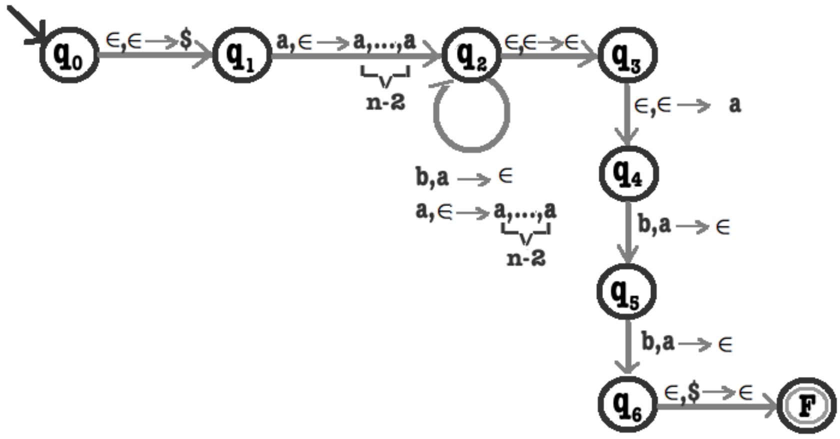

Fig. 5.11: The recognizer (PDA) to the redefined unambiguous Polish Grammar

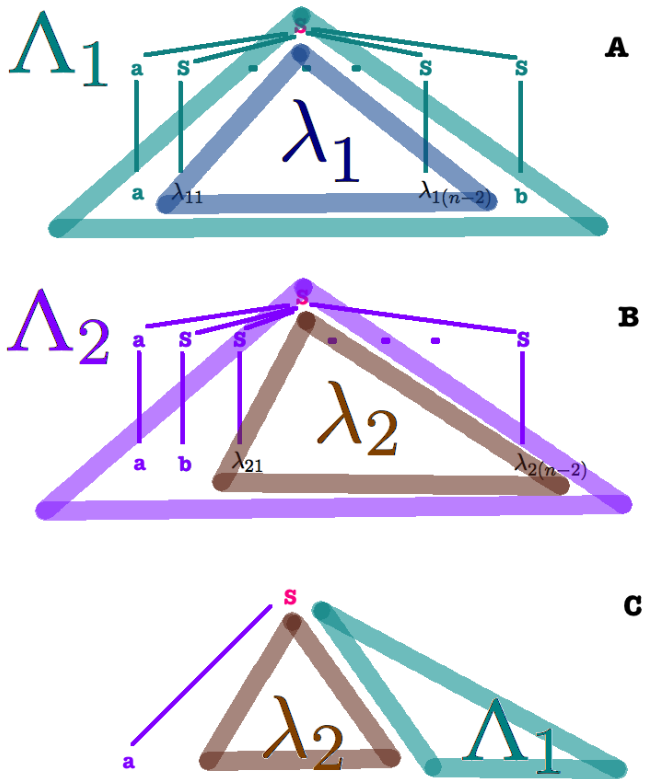

Fig. 5.12: parse Trees -patterns-

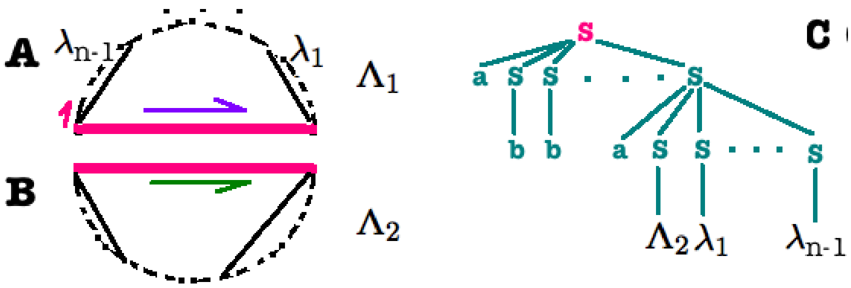

Fig. 5.13: bad-behaved-nGon

5.5 Reinforcing the Universal Flexagon concept

The ultimate flexagon was still explained and exemplified in section 2.1.3, observe that every example corresponds to one triangulated region. To be more accurate, those are the covering space subspaces correlated to the given flexagon maps which was possible due to its nature. However, as it was claimed that any (edge[4]) flexagon map can be derived from the proposed covering space, no matter how many edges the elementary polygon of the map has. At this point, the grammar [section 5.3] to generate maps of any elementary polygon was proposed, it seems appropriate to show some samples having tetra and pentagon as essential polygons.

Attend to figure 5.14.A which is presenting the graph representation of the flexagon map alongside one of its corresponding dual, the Tukey Tetra Network. 5.14.B illustrates the polygonal form which represents the normal form of the flexagon map graph as well as its dual following the same idea of connecting the middle points of the sides. The correlated dual to the normal form along with the addressing of the former will enable us to easily collect the number sequence to generate its frieze code, the previous step to derive the frieze. Finally, 5.14.C reveals the covering subspace correlated with the flexagon map.

The next example 5.15.A illustrates not only the graph of the flexagon map and its Tukey Network but the zero-side and the addressing indicating how this map was generated based on the grammar declared in 5.1 as well the virtual and real sides. In 5.15.B the normalized graph of the polygonal form along with vertices and sides of the corresponding dual to execute the go through the map. Vertices of the normalized form were removed to better appreciate the dual. The subspace based on the universal flexagon is drawn in 5.15.C.

Let us move on to a pentagon example 5.16.B, the parse three is accompanying the graph of the flexagon map [5.16.B] which is including the Tukey Penta Network, the signaling of how the map was generated in accordance with the grammar presented in 5.1. This time the normalized graph and its dual [5.16.C] both were tagged conveniently. Note the shape of the subspace correlated with a chain penta map illustrated in figure 5.16.D.

A penta star flexagon map [5.17.A] could have not been left aside due to it, being slightly more complex but still clear example, reveals the characteristic normalized graph form [5.17.B] of the recursive definition of a map flexagon in which the reverse in orientation is visible on the Tuckerman traverse with each extension to the flexagon (map). In this example, it is more visible how the distribution of the subspace [5.17.C] is correlated with the flexagon map and to its normalized form.

The present chapter finishes with this short sample to add clarification on how the elements revised along the whole work are correlated, flexagon maps, graph representation and its dual, normalization, the Tukey Network, addressing, the Tuckerman traverse, the grammar context,

[4]Consider that Conrad 'n Hartline contemplated as flexagons those that nowadays have been termed as edge flexagons

the zero-side, virtual and real sides, joining flexagons maps to extend them and, the Universal Flexagon.

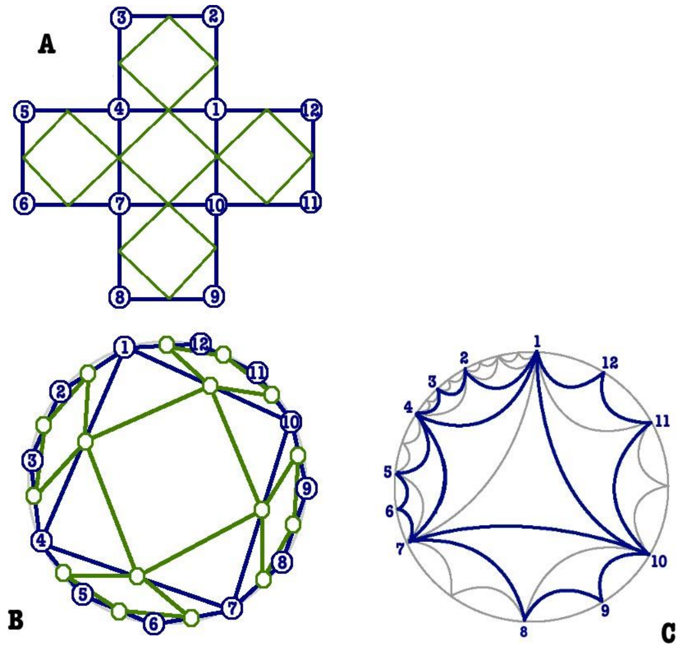

Fig. 5.14: Tetra-to-UF-cc

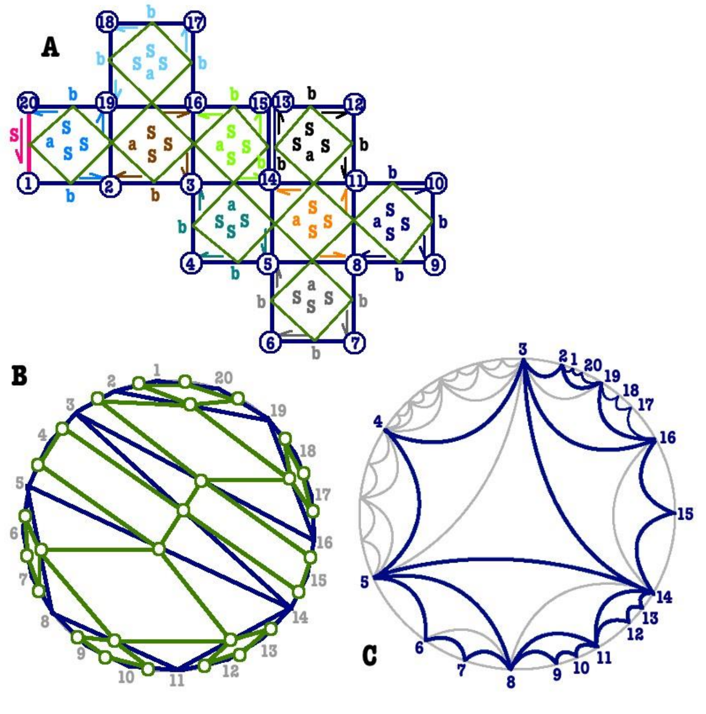

Fig. 5.15: Tetra-grammar-c-UF-v

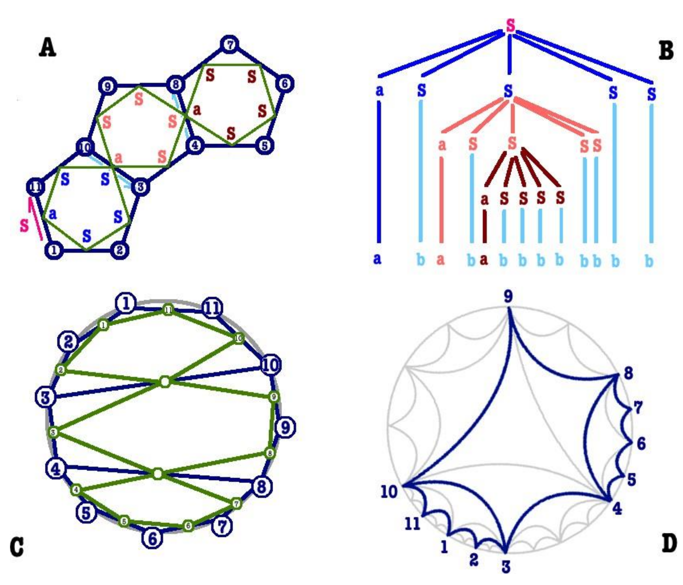

Fig. 5.16: Penta-grammar-i-UF-ii

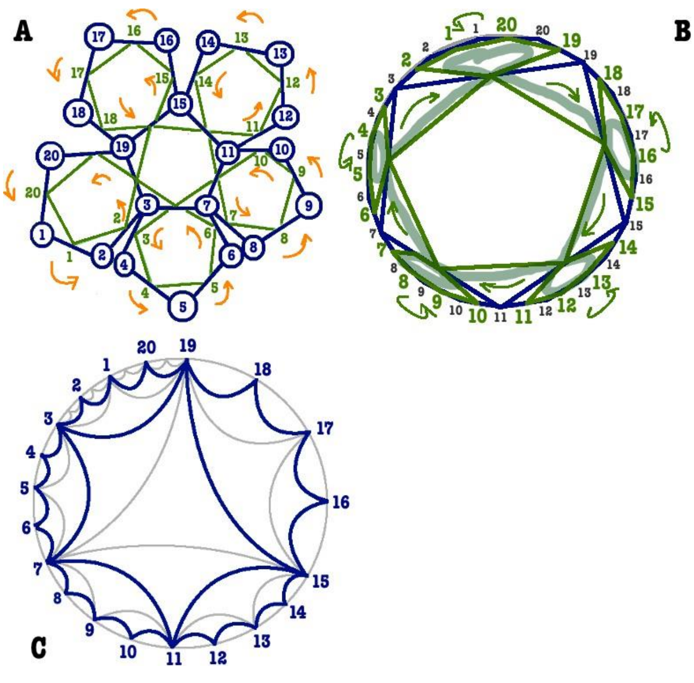

Fig. 5.17: Ex-TukeyPentaNetwork-iA-c-SEFUsii-tt-i

Chapter 6

Conclusions

Summing up the contributions in this document, flexagons were correlated with language theory using the unambiguous Polish grammar [Gross and Lentin., 1967] as well as an extension of it was established. It was showed that proper flexagons are closed under joining operation supported in regular expression's closure property. How the frieze code [Conrad and Hartline, 1962] is obtained was stepped. The McIntosh Method [McIntosh, 2011] was introduced. The Universal Flexagon concept was reinforced exhibiting a few examples of the flexagon maps derived from it.

Mostly Wheeler [Wheeler., 1958], Oakley and Wisner [Oakley and Wisner., 1957], Conrad [Conrad., 1960] and Hartline [Conrad and Hartline, 1962], McIntosh [McIntosh, 1960], [McIntosh, 2000a], [McIntosh, 2000b] Pook [Pook, 2003], [Pook, 2009] and, the flexagon's lovers yahoo group [Schwartz, 2005] give us the basis to get familiarity with flexagons. Wheeler introduced the extension of flexagons presenting circled flexagon maps as predicting that the number of sides of the elementary polygons will become irrelevant. Oakley and Wisner presented their analysis to count flexagons (maps). [Conrad., 1960], [Conrad and Hartline, 1962] and [Pook, 2003] are the printed references to be familiar with properties and attributes of flexagons. As on of the titles of these works [McIntosh, 2000a], [McIntosh, 2000b], they result a quick (short) reference to have a the general context of flexagons.

Conrad and Hartline started to manipulate flexagons in graphs although not in that jargon[a]. McIntosh [McIntosh, 2011] analyzed different possibilities to adequate flexagons maps representation under graph context, looking for the most convenient way to make connecting them in pairs possible. Flexagons have been worked on for fun or involved in more deep analysis, members of the flexagon's lovers yahoo group follow both perspectives.

[a]Bear in mind that they work was presented at the early 60's.

Among a vast variety of bibliography, to have the basis in language theory we referred to [V.Aho and Ullman., 1972]. We follow the steps of Lentin [Gross and Lentin., 1967] and Gross and those of Carroll and Long [Carroll and Long, 1989] to have the elements to have the assertion of this work, attaching flexagon maps in pairs will give us a new flexagon map, first it was showed to triangulated convex polygons, then it was proposed to elementary polygons of n sides. The notes about graphs were predominantly based on [Whitney, 1931] and [Konig, 1990].

Triangulations in polygons considered as flexagon maps are different. Fixing the tags to the sides once the triangulated polygon was generated, rotation and reflection do not affect the parse tree correlated to it. Taking a pair of partitions, forgetting for one second labels and orientations, one achievable from the other with rotation or reflection. Once sides are identically disposed, the parse tree are generated in accordance with tags and orientation, the words produced are different to each other under grammar terms.

Such fact supports the statement that the Catalan Numbers' expression counts the set's elements of different potential flexagon maps given a convex polygon. Knowing that i) Catalan Numbers express the total possibilities in which a convex area can be triangulated, ii) the unambiguous Polish grammar 3.2.2 (3.1) enable us to generate any triangulated convex polygon, iii) flexagon maps can be disposed in a polygonal system, iv) flexagon maps and triangulated polygons can be represented by graphs. Combining all these elements helps us to show how Catalan numbers solve the posed question.

Placing flexagon maps in terms of graphs give us the tool to conveniently manipulate them to go straightforward to its polygonal representation. The normal form which makes not only a clear joining of the corresponding graphs representing a pair of flexagon maps possible but a fluently collection of the number sequence to generate the frieze code of the flexagon extension. Be mindful of the fact that joining, attaching, merging up and, linking terms are handled with the same meaning when joining operation is alluded to.

The sequence number is encoded into the frieze code, whose arrays of numbers combined with the sign sequence are the abstract representation of the pattern. It is possible go back and forth from one representation to another, the extension of a flexagon is reflected as it corresponds in each one. The frieze code is condensing essential data from which the frieze can be derived from and in turn the physical flexagon can be assembled.

The McIntosh method enabled us to extend a flexagon map attaching them in pairs, facilitating the procedure to have the frieze code of the joining result from which the new frieze can be directly built. The opened question was if from the joining product could still be assembled a flexagon. To answer it, flexagons were put in the theory language context using grammars to support flexagon joining closure on the closure property of substitution operation working on context-free languages. In addition, the McIntosh Method introduced to triangulated surfaces playing as flexagon maps, still works on n-gons, the only condition held is that the totality of their vertices stay on the polygon border.

With the original unambiguous Polish grammar 3.1 definition, the number of sides of the polygons disposed to join them is irrelevant. This grammar is able to generate any triangulated convex polygon, in the joining as it has been illustrated in figure 5.2 what it is pertinent to consider to re-set the new virtual starting side is the disposition to the triangles directly involving in the joining not the number of sides of each given polygon, i.e. we can link an n_1-polygon to one n_2-polygon.

After grammar was redefined [section 5.3], not only it is possible to attach n-polygons to m-polygons but there is nothing preventing merging a pair of flexagon maps formed of mixed elementary polygons themselves, i.e. the polygons forming the flexagon map can have a different number of sides. The nature of the definition of the grammar 5.1 supports this declaration.

It is essential to respect orientation in which flexagon maps are linked. This not only will allow you to define in an ordered way the substitution's rules but the most relevant matter is that the joining of the given flexagon maps will be obtained otherwise you can have a flexagon map but not the result of the proposed attaching.

Correctly shifting the starting virtual side is basic to handle joining flexagon maps under substitution operation. Once more, here the relevant thing is to have the joining of the pair of flexagon maps given initially. As part of the process to show closure property in flexagons is the identification of the sub-trees which has been completed proposing the possible sides to make the shifting.

Restricting the re-assignation of the virtual side to the immediate side next to the original one with respect to the addressing, gradually lost relevance once you acquire familiarity and the suitable substitution rules can be defined. However the options you have to redefine the new one are limited to the "free" sides of the elementary polygon directly implicated in the linking.

Contextualizing flexagon maps in terms of graphs and grammars, it has been possible to show that the result of joining them in pairs is again a flexagon map from which we can assemble a new flexagon. The graph theory perspective handled by the McIntosh method[1] combined with the unambiguous Polish grammar 3.1 defined by Lentin and Gross [Gross and Lentin., 1967] enables us to connect triangulated flexagon maps in pairs to obtain a new flexagon map which is supported on the property of closure of the substitution operation.

Understanding how the covering space proposed as the abstract representation of the map symbolizing the Ultimate Flexagon as termed by Conrad 'n Hartline, is correlated to the graph of the flexagon map or even directly with the flexagon map shows why it is delineating the Universal Flexagon, as McIntosh referred to, containing every potential flexagon map formed by elementary polygons of n sides and thanks to the normalizing process in graph context convex polygons are possible. Vertices on the border policy under which flexagons maps are governed is never removed.

[1][Section 4.3]

Appendix A

More flexagons properties and characteristics

The ongoing appendix is dedicated to listing a few more properties which generally characterize flexagons. The vast majority of information is based on [Conrad and Hartline, 1962]. Here the material was condensed, revise Conrad and Hartline [Conrad and Hartline, 1962] paper for detail. The Pook' published books [Pook, 2003], [Pook, 2009] include part of these results which were rediscovered by him.

Immediately it can be noted that symmetry and stability attributes are compromised (undermined) when not all foldings, in process of assembled, follow the same orientation. Flexagons can be identified as a Twist Moebius band, specifically the chain type, which are characterized by triple symmetry. When the elemental polygons used have four or more sides will imply bi-symmetry, this means that at least a pair of plans are needed in order to assemble a flexagon.

While assembling, unexceptionally, pairs of contiguous frieze leaves crease together. When it is completed, the pile of the flexagon is immutable under pinch[1] flexing. The routes in the "go-through" map are a guide to reveal any given side one pleases. It does not admit triangle vertices completely enclosed. The arrangement of triangles, the map, satisfying cited restrictions determines uniquely any flexagon. Obviously an interdependence between the map and the plan is held. But noted that different maps can share the same plan.

If single thickness of a pat is achieved, the only option to continue flexing is after rotating. The unique alternative when a flexagon extension is avoided. A sign sequence composed entirely by pluses can be replaced by minuses and conversely due to positive sense of a map is set in a subjective way. The course of the path described by the Tukey triangle network strictly traces the Tuckerman traverse. There is a bijective relation between Tukey network lines and the hinges in the unit.

A process of reduction on a sign sequence helps to identify if it corresponds or not to a flexagon. Given a flexagon sign sequence, opposite systematic reduction from several cycles to only one is feasible. It is done exchanging G - 1 consecutive pluses or minuses by merely a contrary one. In the opposite, the result of adding signs contained in a given sign sequence is congruent to 0 mod. G. Each flexagon map can be characterized as a composition of the generally identified flexagons's classes: the chain, the fan and the star [Conrad and Hartline, 1962], each one having a featured sign sequence.

There is another classification based on a flexagon methodical treatment [Pook, 2003]: the convex polygon and star flexagon families. Although a similar behavior is identified between the star and its correlated convex polygon flexagon. The same plan[2] can give place, depending on

[1]Although original source [Conrad and Hartline, 1962] did not specify a kind of flexing, it is explicit noted here because nowadays there are different sort [Sherman, 2000] of them.

[2]Pattern, frieze

folding, to more than one flexagon. A bijective relationship is recognized between each flexagon side and each point along the contour of its map. If a pair of cycles constituted by constant orders are equal, then the flexagons related with each one may be considered identical.

Operating a flexagon is independent to the variation in the leaves angles. A flexagon of any number of sides, of units and of any polygon shape; is hypothetically conceivable. To some types of flexagons, Pook had identified different main positions [Pook, 2003], which could be understood as the positions in which the flexagon is not in being affected by any type of flexing, i.e., the flexagon is not in transit. No matter the polygon used, the Tukey Triangle System maintains. The map of a cycle *G*-flexagon will be constructed of polygons with *G* sides, joined to another by single sides. Each flexagon contains veiled tubulations between non-adjacent sides of a particular cycle. The first hinge is used for forward movement and the one before the end to go towards the back.

Labeling thumbholes 123 .. .$(m - 1)$ from up to down. It is observed that number 1 hinge is related to thumbhole 1 and used in a 1-flex. Thumbhole n will be related to n hinge and used in n-flex. The unit of simple proper flexagons maintains relation with a pair of hinges and every hinge with a pair of leaves. The hinge addressing establishes a base of reference of an unfolded flexagon. The hinge difference is autonomous of where the zero point is set in the hinge sequence. The leave hinge holds an invariable correlation with the vectors of that leaf, a consistent pointing towards or away from one leaf to another evidences that.

In general, vectors of cycles sharing one side, except the side in common will be opposite oriented relative to each other. How a flexagon is wound up will define if it is right or left flexing. Reversing the number sequence or the sign sequence make it attainable. Left and right flexing flexagon mirror each other. In a proper flexagon, it will fail to attach a new side to a hidden tubulation. Flexing on a hidden tubulation will skip other sides from the normal cycle. Backward flexes through the omitted sides resulted from transposing two hinges will succeed it immediately, prior to completing the remaining cycle. Hidden tubulation could come from any side to all non-adjacent sides. Evidently there are non improper triflexagons.

As in proper flexagons, identical pat structure, map, and number sequence characterize improper ones. A $+n$-flex is congruent a - $(G - n)$-flex mod.G. The essence of the Polygon System comes from the fact that neighboring cycles have contrary orientation inverting plan signs. A + or - addressing works properly for proper flexagons however due to improper ones handle further than ±1 it is not suitable. Summing up some points related to a given map (to obtain the nature of any tubulation):

- It is feasible to establish a positive orientation then conveniently orient every 1-flex resting on 0-cut cycle.
- Finding an alternative path of 1-flexes crossing strictly the tubulation and connecting them; permits assigning values to each 0-cut higher than 1-flex.
- A +1 hinge difference is obtained whether the 1-flex path advances according to the assigned vector on the map and minus one if they are contraries to each other.
- Total attentive addition of the hinge differences corresponding to 1-flexes matches with hinge difference traveling across the tubulation.
- Generally, adding up the values of the flexes beside 0 - *cut* cycle segment from *A* to *B*, any pair of sides, results in the value of a tubulation from *A* to *B*.

Let us list several diversifications emerged directly from flexagons:

- Without delay, how the angles of the leaves and the angles of the map polygons are related acquires no significance.
- Relevance stays on the number of the sides more than on the shape of the polygons implicated.
- To allow dissimilarities in the angles and in the lengths of the sides don not modify operations as that using regular leaves.
- Some faces becomes unstable because of the use of more units to grant flexibility.
- Setting a as the minimum angle of the leaf, it has been used at least $180^o a$ units to make irregular polygons coincide in the assembled flexagon.

- Irregular polygon edge [Pook, 2009] flexagons inherit partial behavior from the regular one, for instance, their flexing characteristic.
- Stage in which the angles pointed at center of the leaves in turn do not compose 360^o will prevent more flexing. This kind of behavior is represented in the map as a mid-fractured face.
- Given an arbitrary pair of paths resting on contiguous cycles; the midpoint of the common side crossed by the two paths holds broken faces.
- Face degree is independent of the regularity of the leaves.
- The use of irregular leaves may imply that not all hinges are reached.
- To involve nonuniform leaves may affect the number of positions between incoming and outgoing hinges merely deleting some or attaching new ones.
- In compound faces each unit presents 2 values related to the face degree.
- Pats ordering from side ordering should be distinguished.
- Class and face degree are determinants to push through operation.
- The flexagon's state in which it is not possible to execute the push through at a particular face presuppose the impossibility to leave it unless going back across the last incoming side[3].
- In theory, flexing process stays autonomous from the sum of the angles around the center face.
- Every time, adding up signs of the leaves in each single-sub-pat is identical to the face degree.

The map of a 0^o faced flexagon have no changes with respect to those not 0^o faced. And in general it is manufactured with minor adjustments. The 0-cut cycle has absence of 0^o faces. Given a flexagon, the fact of a class being less than its cycle is an indicator that it is 0^o faced.

Heterocyclic property characterizes all incomplete flexagons. The class will do the difference (the only one) between incomplete and general heterocyclic flexagons. The form of the involved polygon will determine the quantity of sides needed to complete the cycle. No matter which dimension of the second cycle is aspired, it is attainable by modifying in the manufacturing of this cycle the sort of the polygon incorporated.

Besides the amalgam of cycles involved, it is possible to manufacture a flexagon maneuvering several polygon plans. Maps composed of mixed polygons are inherit to heterocyclic flexagons. The mechanism in which the flexagon works is not influenced by the number of sides of the polygons utilized. Relative to this fact the class is irrelevant.

Compound faces happen when the aggregation of the sign sequence is not congruent to zero mod 360^o. Alterations in sign sequences affect both, the arrangement of hinges and the contour of the leaves. Property that compound flexagons not situated around the center of the flexagon in an even ring of pats is the dissimilarity's essence between compound and non-compounds flexagons. Requesting more units than other flexagons is the normality to compound ones. Arbitrary shapes and largenesses of the leaves becomes irrelevant. Being n an angle, the arrangement $n, -n, n, -n, \ldots$ is the unique irreducible sign sequence due to that in the process of reducing contiguous terms by its sum (mod 360^o) it is non-zero.

Irregular cycle even edge flexagon and first order fundamental even edge flexagons present the same topological invariants, the different positions identified, as well as, the intermediate position map [Pook, 2009].

[3]This particularity was illustrated in the map as a broken line too [Conrad and Hartline, 1962].

Appendix B

The McIntosh method, more studies on flexagons’ maps representation

The current appendix shows a sample of the varios efforts that McIntosh made to have flexagons abstract representation. Which seems constructed to be optimized to get a clearer softly joining of flexagons in pairs. The nodes seem to be arranged in such a convenient way holding them on the vertices of a quite-regular polygon, except for one resting at the center of it. A second quite-regular polygon is revealed inside the former, once the Tukey Triangle Network is added to.

The original picture showed in B.2 was made up in B.3 to highlight what it is contending, the flexagon map graph (blue edges, pink nodes), Tukerman Traverse (green edges, cyan nodes), Tukerman Tree (black edges, blueberry nodes). Nothing prevents this graph from being the original flexagon map, however, the model’s distribution of the nodes suggests trying to get a polygonal disposition of them. Additionally it is quite similar to the prototype in B.1 in which the distribution of the nodes shows clearly two levels in their alignment. As a parenthesis, note that in this design (B.1) the node eight should be have after the seven was completed, and nine after getting the eight node according to the Tuckerman Traverse.

Although the similarity in the nodes’ distribution was appointed, it should be observed that the disposition of the nodes in B.2 are correlated to the flexagon map and those in the double circled collection (the cyan ones), in B.3, belong to the dual of the flexagon map graph.

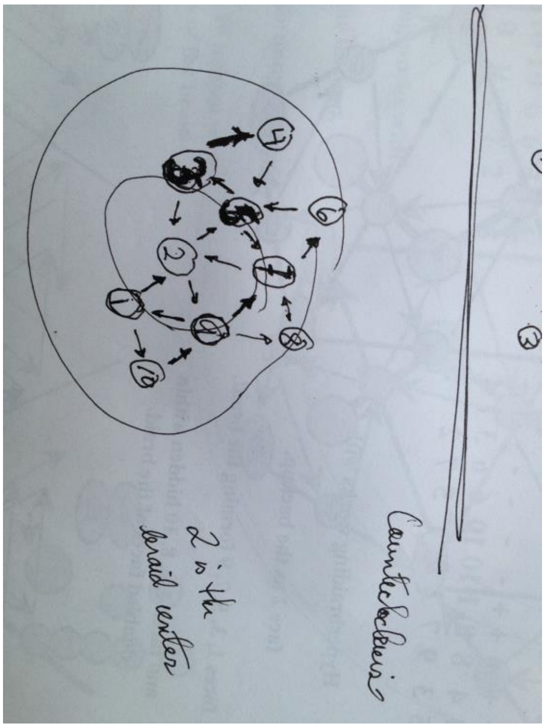

Fig. B.1: FT-graph-paper-SummerXXXIV

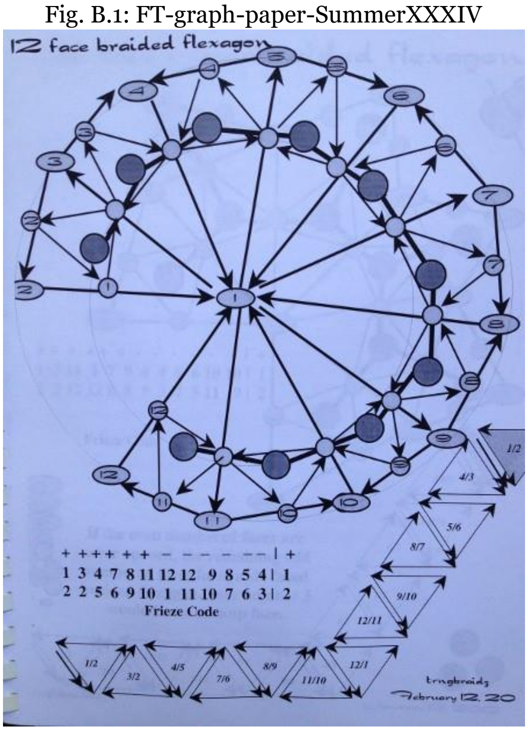

Fig. B.2: 12fbf-SummerXXXIV

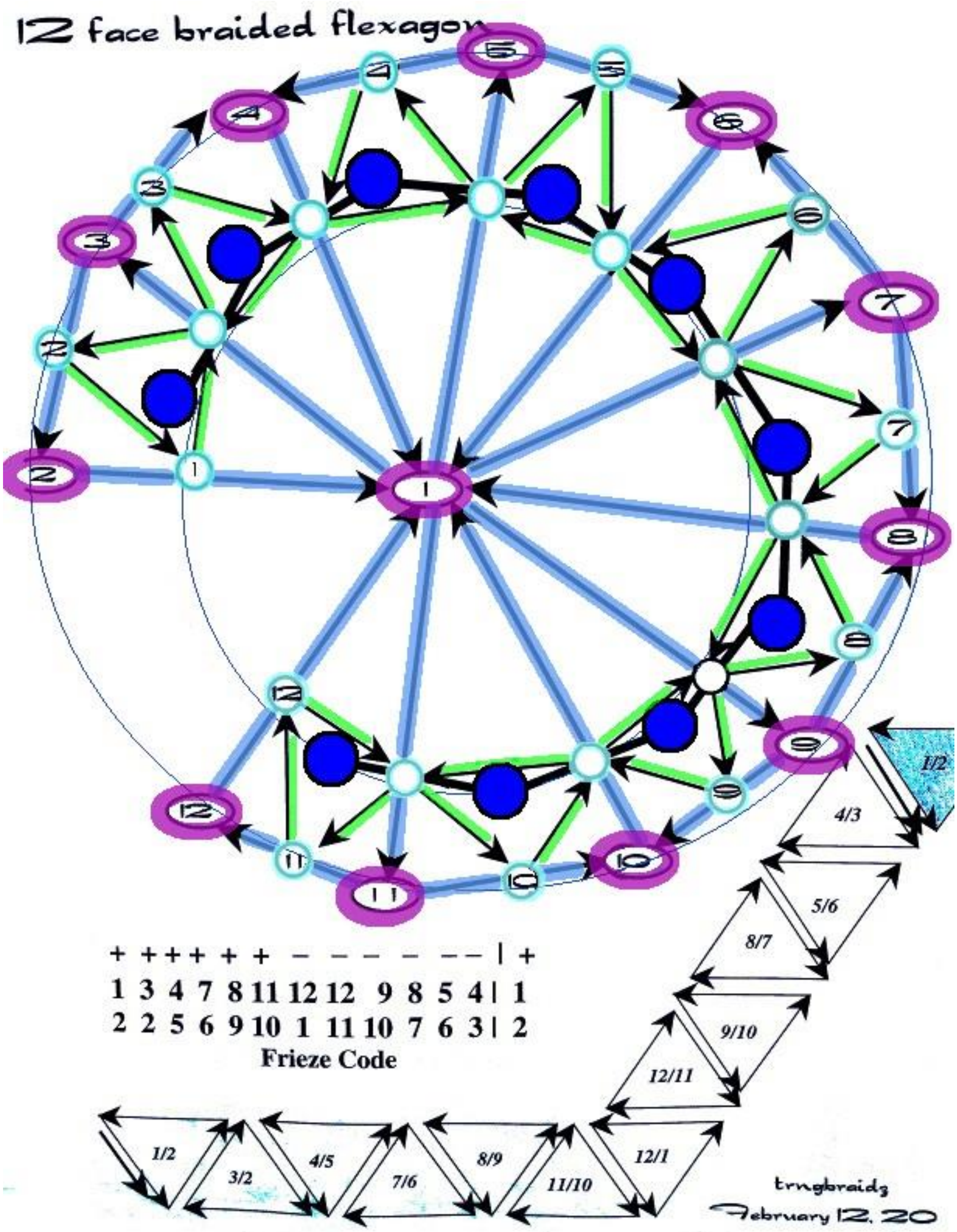

Fig. B.3: 12fbf-SummerXXXIV-rtd

Appendix C

Explicit parse trees

The full detail of the parse trees presented here is completing the samples showed in the chapter 5.

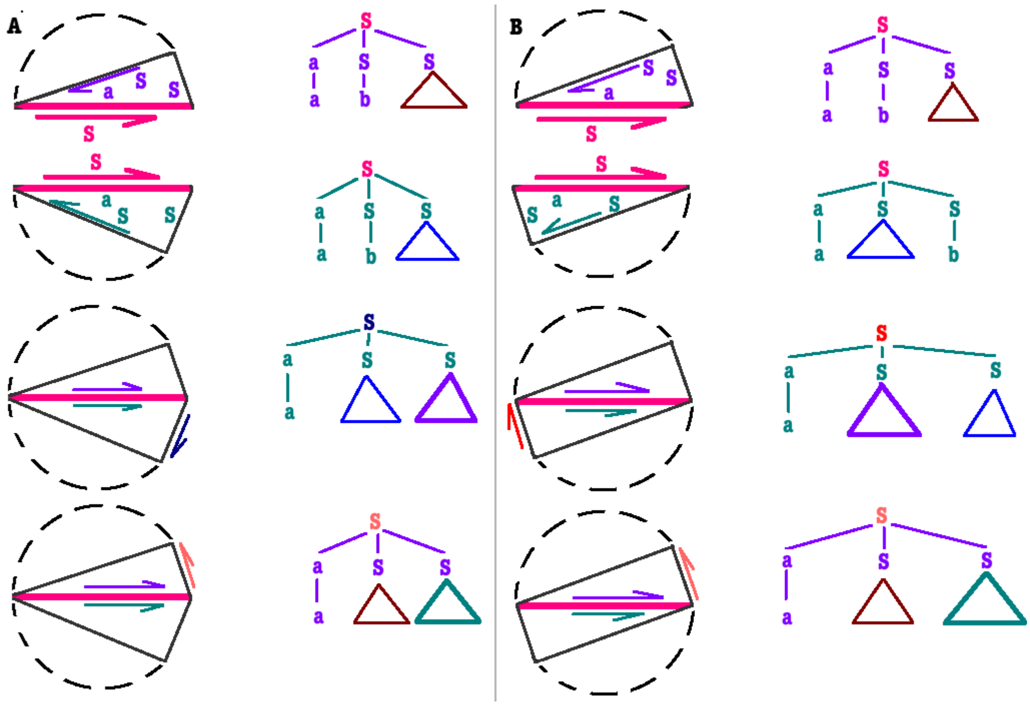

Fig. C.1: joinPosTree-i3-4l

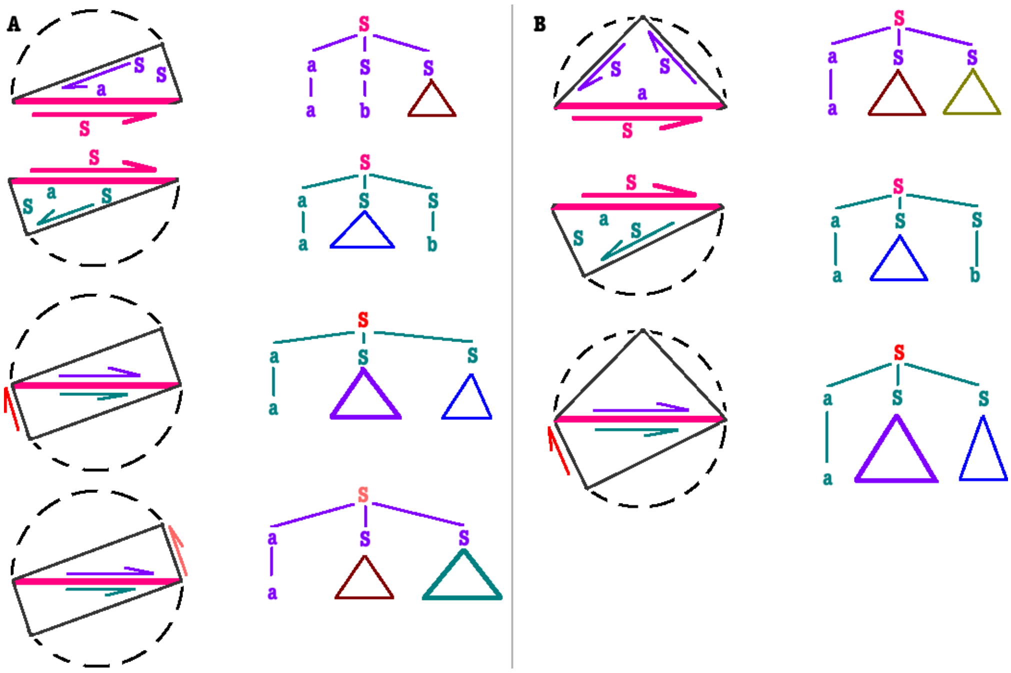

Fig. C.2:joinPosTree-i5-6l

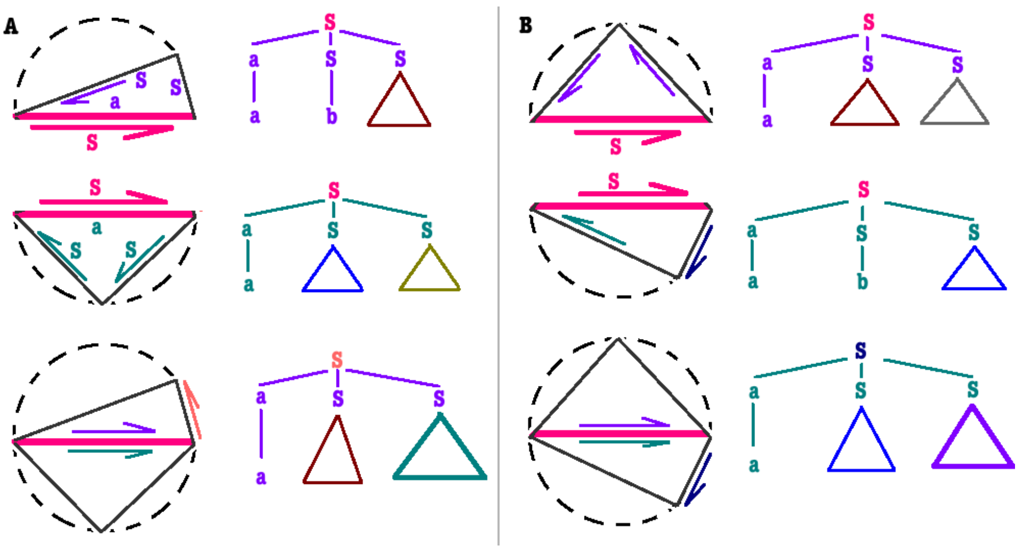

Fig. C.3:joinPosTree-i7-8l

Appendix D

One more example

It is convenient to illustrate one more example, although some steps were not explicitly showed, here, the idea is to exhibit how things happen in the original flexagon maps when they are merged. The alphabets $\sum = \{a, b\}$ and $\sum' = \Lambda_1, \Lambda_2, \lambda_1, \lambda_2, \lambda_3, \lambda_4$ the same determined in 5. With this example, it could be appreciated that we have two possible candidates to shift the starting virtual after the flexagon map linking.

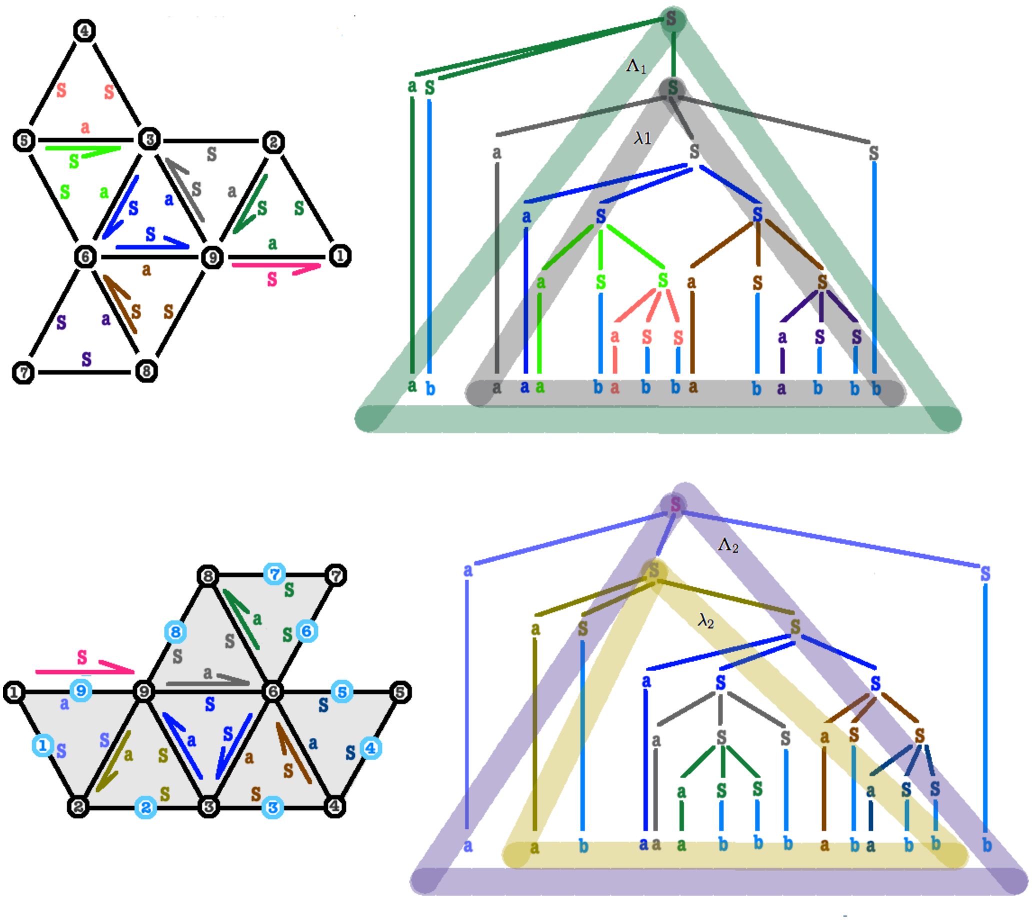

Fig. D.1: torship-g

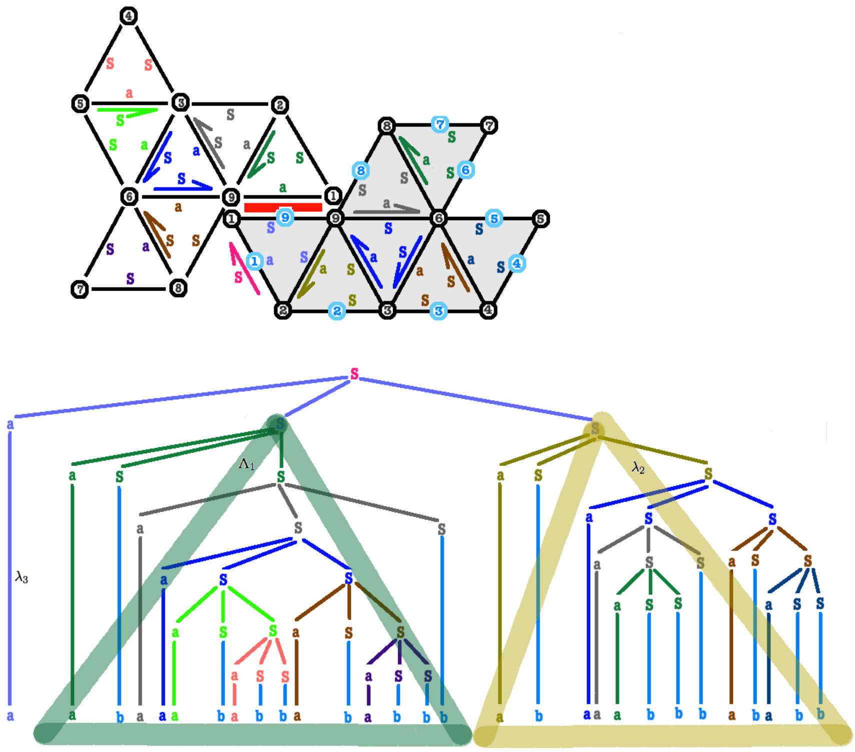

Fig. D.2: TS-joinGram-ii-jcos

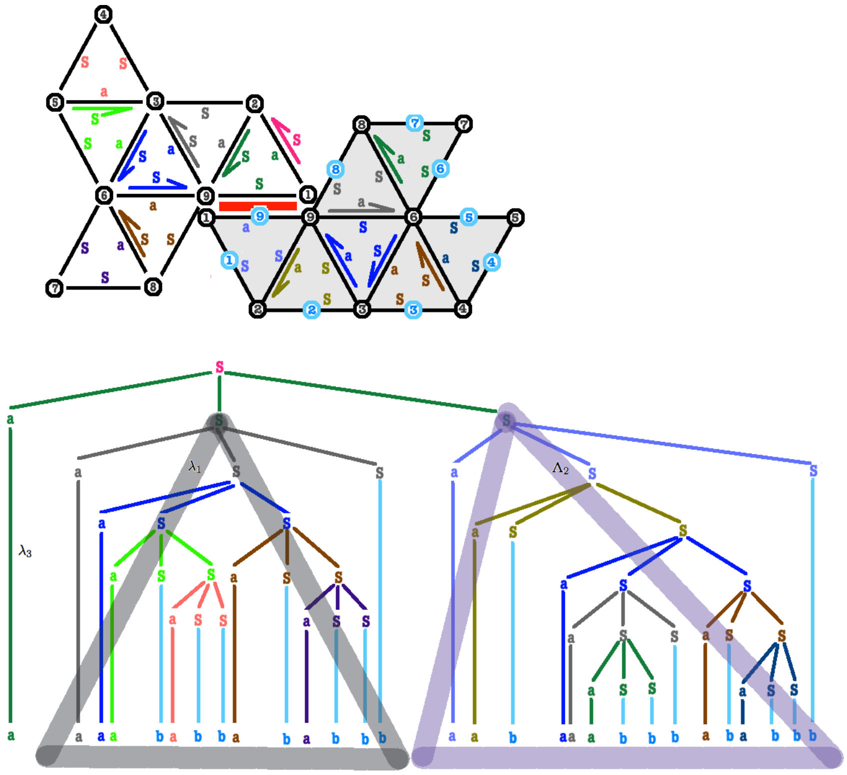

Fig. D.3: TS-joinGram-ii-css-2

Bibliography

[Amarasinghe et al., 2015] Amarasinghe, S., Chlipala, A., Devadas, S., Ernst, M., Goldman, M., Guttag, J., Jackson, D., Miller, R., Rinard, M., and Solar-Lezama, A. (2015). Reading 17: Regular expressions & grammars. http://web.mit.edu/6.005/www/fa15/classes/ 17-regex-grammars/. [Online; accessed 29-July-2018].

[Andrásfai, 1977] Andrásfai, B. (1977). Introductory Graph Theory. Pergamon Press.

[Barcellos and Gardner., 1979] Barcellos, A. and Gardner., M. (1979). A conversation with martin gardner. In The Two-Year College Mathematics Journal, volume Vol. 10, No. 4, pages pp. 233–244. Mathematical Association of America.

[Batista et al., 1999] Batista, G. D., Eades, P., Tamassia, R., and Tollis, I. G. (1999). Graph Drawing, Algorithms for the Visualization of Graphs. Prentice Hall.

[Behzad and Chartrand, 1971] Behzad, M. and Chartrand, G. (1971). Introduction to the Theory of Graph. Allyn and Bacon, Inc.

[Bollobás, 2000] Bollobás, B. (2000). Modern graph theory. In Graduate Text in Mathematics, volume Vol.184. Springer.

[Bondy and Murty, 2008] Bondy, J. and Murty, U. (2008). Graph Theory, Graduate Text in Mathematics. Springer.

[Buser et al., 1994] Buser, P., Conway, J., Doyle, P., and Semmler., K.-D. (1994). Some planar isospectral domains. International Mathematics Research Notices, 1994(EPFL-Article-161424):391–400.

[Carroll and Long, 1989] Carroll, J. and Long, D. (1989). Theory Of Finite Automata With An Introduction To Formal Languages. Prentice Hall.

[Charles.L.Hopkins., 1961] Charles.L.Hopkins. (1961). Polymorphic geometrical devices. https://www.google.com/patents/US2992829. Patent number 2,992,829, Serial No. 603,062. Issued July 18, 1961. Filed Aug. 9, 1956, Chicago III (3173 Strathmore Drive, Ventura, Calif.).

[Conrad., 1960] Conrad., A. S. (1960). The theory of the flexagon. Technical Report Technical Report 60-24, RIAS Miscellaneous Publication.

[Conrad and Hartline, 1962] Conrad, A. S. and Hartline, D. K. (1962). Flexagons. Technical Report TR 62-11, RIAS, 7212 Bellona Avenue Baltimore 12, Maryland.

[Dershowitz and Jouannaud, 1990] Dershowitz, N. and Jouannaud, J.-P. (1990). Rewrite systems. In Handbook of Theoretical Computer Science, volume Volume B: Formal Methods and Semantic, pages pp. 243–320. Mathematical Association of America.

[Diestel, 2000] Diestel, R. (2000). Graph Theory. Springer-Verlag.

[(G4G), 2012] (G4G), F. G. . G. (2012). gathering4gardner. http://www.gathering4gardner.org/. [Online; accessed 28-February-2018].

[Gibbons, 1985] Gibbons, A. (1985). Algorithmic Graph Theory. Cambridge University.