Rigid Motions of K_5 Obtained from D_6 Action on Planar Rooted Trees

Sebastian Karlsson
sebastiankarlsson29@gmail.com

under the direction of
Dr. Benjamin Ward
Department of Mathematics
Stockholm University

Research Academy for Young Scientists
July 11, 2018

Abstract

This study investigates associahedra - polytopes which are interesting in various mathematical disciplines such as topology and combinatorics. We prove that the D_6 action on the planar rooted trees correspond to rigid motions of a certain realization of K_5. We also offer an explicit description of which element in D_6 corresponds to what type of rigid motion of K_5. In further research this could be generalized to K_n.

Acknowledgements

I would like to express my gratitude towards my mentor Dr. Benjamin Ward for putting so much time and effort into helping me and for coming up with very interesting research questions. I would also like to thank my collaborator Ellinor Jönson who has played a large and significant role in this study. I would like to thank Rays – for excellence and their partners Svenska Tekniska Vetenskapsakademin i Finland and Kjell och Märta Beijers Stiftelse.

Contents

1 Introduction

The mathematical structures called associahedra of dimension n, denoted K_{n+2}, is a certain kind of polytope in n-dimensional space. It was first described in 1951 by the mathematician Dov Tamari in his doctoral thesis [1]. The associahedron has been used to investigate the property of associativity in topology by among others, J. Stasheff [2]. The structure is interesting in many areas of mathematics, e.g. in the theory of operads and homotopy theory.

The associahedron is closely related to combinatorics since its vertices are in one-to-one correspondence with a type of graph called planar rooted binary trees. These graphs have symmetry and this symmetry can be described with a mathematical structure called a group. The aim of this article is to describe how these symmetries can be seen on the associahedron.

The symmetries of the associahedra has been studied on an abstract level by Carl W. Lee [3]. We offer a more hands on and explicit presentation.

2 Basic Definitions and Examples

2.1 Basic Concepts in Group Theory

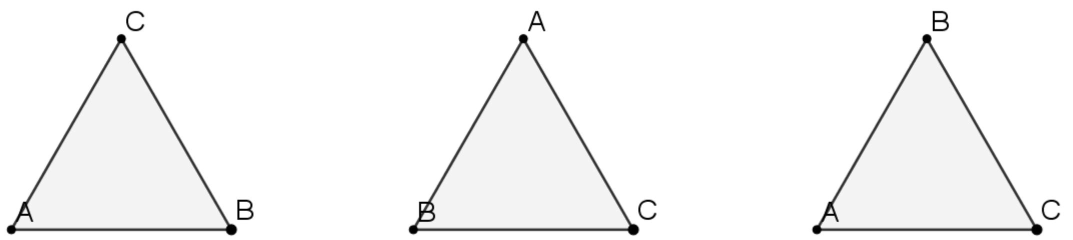
We begin by giving definitions of some basic concepts necessary for understanding the investigation of the symmetric properties of the associahedra. Symmetry is described by a mathematical concept called a *group*.

Definition 1 (Group)**.** A group G is a set together with an operation $\oslash$ which takes any two elements of G and produces a third. The operation should be associative, i.e. for all $a, b, c \in G$, $a \oslash (b \oslash c) = (a \oslash b) \oslash c$. There should exist an element e such that $e \oslash a = a \oslash e = a$. Lastly, every element in G should have an inverse element, that is, for all $a \in G$ there exist $\acute{a} \in G$ such that $a \oslash \acute{a} = \acute{a} \oslash a = e$.

The element e is called the identity element. Note that it is not necessary that $a \oslash b =$

$b \oslash a$ for all $a, b \in G$. If that is the case, then the group is an *abelian* group. Often, $a \oslash b$ is just written as ab and $\underbrace{a \oslash a \oslash \ldots \oslash a}_{n \text{ times}}$ as a^n. It is easy to show the uniqueness of the identity element and the inverse element of an element a. We give a basic example in order to make the concept of a group more easy to understand.

Example 1. The dihedral group D_n is the group consisting of all rotations and reflections of a regular polygon of n sides. D_3 is the rotations and reflections of a triangle and it consists of 6 elements. The identity element leaves the triangle unchanged[1]. If one labels



(a) The initial configuration of the triangle and its labels.

(b) Rotation of Figure 1a by $4\pi/3$ radians as dictated by ρ^2.

(c) Reflection across the line perpendicular to BC in Figure 1a.

Figure 1: Illustration of how the elements of D_3 affects a triangle with labeled vertices.

the vertices of the triangle as in Figure 1a, then a rotation of $4\pi/3$ radians produces the triangle in Figure 1b. That rotation is denoted ρ^2 since it can be seen as the initial rotation ρ of $2\pi/3$ radians applied twice. Furthermore, a reflection along a line perpendicular to BC in Figure 1a, produces the triangle in Figure 1c. The group operation $a \oslash b$ could be seen as first rotating or reflecting the triangle in the way b describes, then rotating or reflecting the triangle in the way a describes.

A *subgroup* is a subset of the elements of a group such that the subset itself forms a group under the same operation. The *order* of a group is the number of elements a group contains. The order of a subgroup must always divide the order of the group. This is a central result in elementary group theory known as Lagrange's theorem.

[1]Can be seen as rotating the triangle three times.

Definition 2 (Group action)**.** Consider a non-empty set X and a group G, a *group action* on the set X is a function $\theta : G \times X \to X$. Let e be the identity element of G and $g, h \in G$. Let $x \in X$. The group action should satisfy the following: $\theta(e, x) = x$ and $\theta(g, \theta(h, x)) = \theta(gh, x)$.

The triangles in Figure 1 can be viewed as elements in a set which D_3 acts upon.

The *generators* of a group G is any set of elements of G such that all other elements in G can be viewed as the generators composed with each other in some way. Additionally, among the generators there should not exist some element which can be viewed as the other generators composed with each other.

Definition 3 (Orbit)**.** The orbit of an element x in a set X under the action of a group G is the set $\{\theta(g, x) : g \in G\}$.

In this paper we will consider action on sets by the dihedral group. When talking about orbits, we will mostly mean the set $\{\theta(g, x) : g \text{ is a generator of } G\}$. For the one interested in groups or would like to know more, we recommend the reader to pic up any book on abstract algebra such as *Contemporary Abstract Algebra* by J. A. Gallian.

2.2 Basic Concepts in Graph Theory

A *multiset* is a set-like structure in which duplicates of elements are allowed but no order is defined between the elements. A multiset can be denoted by square-brackets. We now define *graphs* - a central concept in combinatorics.

Definition 4 (Graph)**.** A graph, H, is an ordered pair $H = (V, E)$ of a set V, called the *vertices* and a multiset of pairs of vertices, E, called the *edges*. If the pairs of the vertices are ordered pairs, then the graph is an *ordered* graph. If the pairs of the vertices are unordered, then the graph is an *unordered* graph.

In this article a graph - if not explicitly pointed out - is an unordered graph. Note that in combinatorics, a graph is not a way of representing functions, but a way of describing a relation between things. The "things" are visualized by dots, called nodes or vertices, and the relations are visualized by connecting two nodes with an edge.

Example 2. Let H be a graph, $H = (V, E)$, where $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$ and $E = [\{v_1, v_2\}, \{v_1, v_2\}, \{v_1, v_2\}, \{v_2, v_3\}, \{v_2, v_4\}, \{v_3, v_4\}, \{v_4, v_5\}, \{v_5, v_6\}, \{v_3, v_3\}]$. Then the graph is represented by:

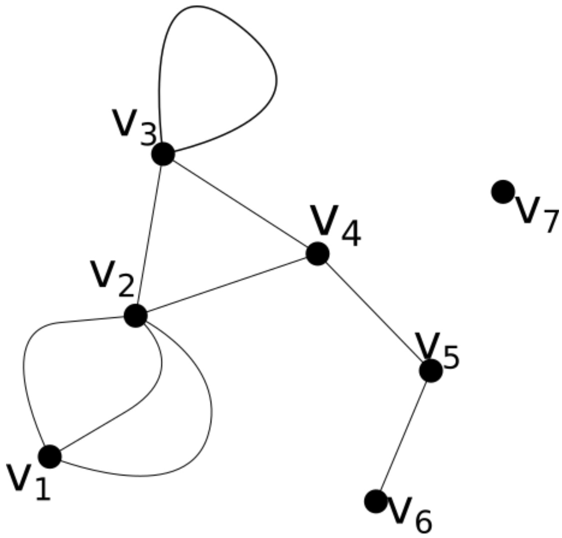


Figure 2: An example of a graph.

We now move on to define the type of graphs we will work with in this paper.

Definition 5 (Tree)**.** A *tree* is a graph in which any two vertices are connected by a unique path.

The graph in Figure 2 is not a tree since not every vertex is connected to another one by a unique path.

Definition 6 (Planar rooted tree, PR-tree)**.** A tree is a *planar rooted tree* if one vertex is marked and the tree is embedded in the plane. In this paper we add the additional restriction that no vertex are allowed to have *degree* two. The degree of a vertex is the number of edges which contains that vertex. The marked vertex is called the *root* and should have degree one. All other vertices with degree one are called *leaves*. The vertices with degree one are called *outer vertices* and all other vertices are called *inner vertices*.

Definition 7 (Planar rooted binary tree, PRB-tree)**.** A *planar rooted binary tree* is a planar rooted tree with no vertex of degree greater than three.

2.3 Polytopes and Their Combinatorial Interpretation

A *polytope* is a generalization of a polygon or a polyhedron to any dimension. Basically, an n-polytope can be seen as a shape in n-dimensional space with "flat" sides. This intuitive notion is made mathematical by the notion of the *convex hull* of a set of non-coplanar points in $\mathbb{R}^n$.

A k-*cell* is a k-polytope which is a part of an n-polytope, e.g. the 0-cells of a cube are its vertices, the 1-cells of a cube are its edges and the 2-cells of a cube are the squares making up the faces of the cube. The 3-cell is the cube in itself.

Example 3. The simplest kind of polytope is a *simplex*. A simplex of dimension n is the convex hull of $(n+1)$ non-coplanar points in $\mathbb{R}^n$. The intuitive notion of a simplex is that it is an n-dimensional triangle. A simplex of dimension 2 is any triangle and a simplex of dimension 3 is any tetrahedron.

The combinatorial structure of a polytope is the way different cells are connected to each other, i.e. the size and proportions are irrelevant. The combinatorial structure of a polytope can be described by something called a *face poset*. In this article we define polytopes in terms of their combinatorial features and by a *realization* we mean a way to describe the polytope geometrically.

Definition 8 (Partially ordered set)**.** A partially ordered set, also called *poset*, is a set, S, together with a relation, $\preceq$, such that the relation is defined between certain elements $a, b \in S$ but not necessarily between all elements. The relation should satisfy the following: for any element $a \in S$, $a \preceq a$. If $a \preceq b$ and $b \preceq a$ then $a = b$. Lastly, if $a \preceq b$ and $b \preceq c$ then $a \preceq c$.

Example 4. A partially ordered set can be visualized with a directed graph called a *Hasse diagram*. This is done in Figure 3. In a Hasse diagram, for an element a to be placed on a higher level than another element b, means that $b \preceq a$. The edges in a Hasse diagram illustrate which vertices the order relation is defined between.

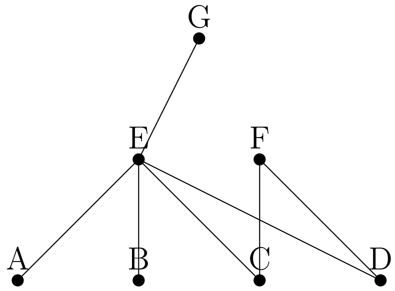


Figure 3: A poset visualized with a Hasse diagram. A $\preceq$ E $\preceq$ G is true but F $\preceq$ G is not defined.

Definition 9 (Face poset)**.** A face poset of a polytope is the poset of all cells ordered such that A $\preceq$ B if A is contained in B, i.e. A is a cell and a subset of B.

Example 5. Consider a cube with labeled vertices as in Figure 4. Then the face poset is visualized with its Hasse diagram in Figure 5.

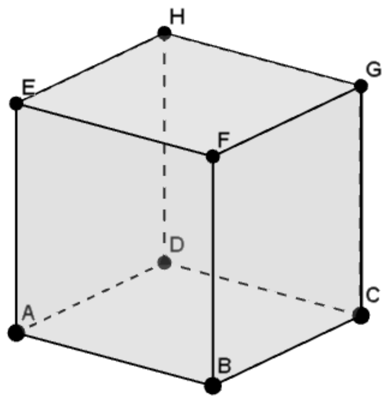


Figure 4: A cube with labeled vertices. The other cells are labeled by what vertices they contain. The face poset consists of all cells ordered such that any cell is less than another if that cell is contained in the other, e.g. A $\preceq$ AB $\preceq$ ABEF $\preceq$ ABCDEFGH.

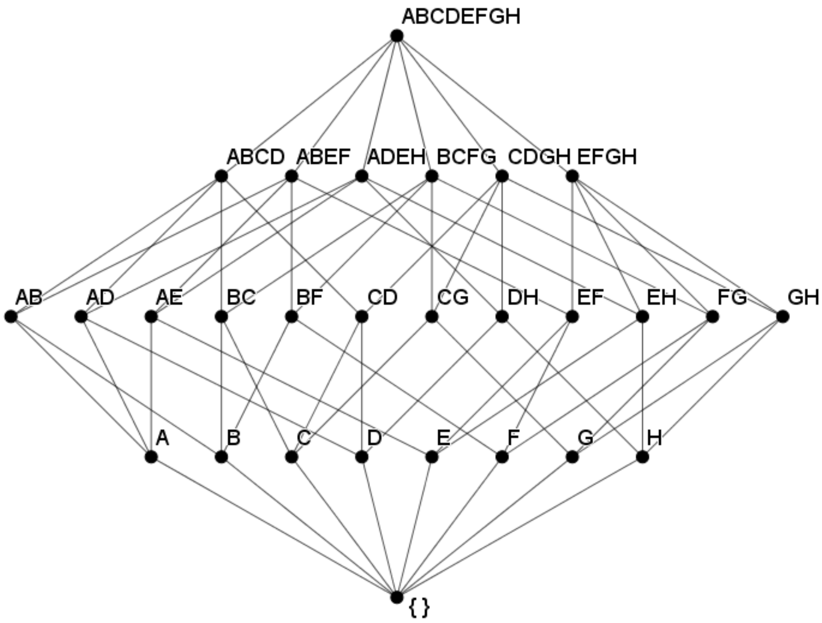


Figure 5: Hasse diagram representing the face poset of the cube in Figure 4. Since the empty set is a subset of any set, the empty set is sometimes included in the face poset.

Intuitively, two posets are *isomorphic* when they are structurally the same. The formal definition can be found below.

Definition 10 (Isomorphic posets)**.** Two partially ordered sets X and Y with order relation $\preceq$ and $\trianglelefteq$ respectively, are considered to be isomorphic if there is a one-to-one correspondence $f : X \to Y$ such that $x_1, x_2 \in X$ and $x_1 \preceq x_2$ if and only if $f(x_1) \trianglelefteq f(x_2)$.

The map f is said to be an *isomorphism.*

3 The Associahedra

The Associahedra are a family of polytopes of which there exists one in every dimension. The polytope is defined entirely by its combinatorial structure. However, the simplest way to define what associahedra is requires an understanding of something called *diagonalized polygons.*

3.1 Diagonalized Polygons

A regular polygon in which 0 or more diagonals are drawn between any two non-consecutive vertices in such a way that no diagonal cross another diagonal is called a diagonalized polygon. In this article, we denote the set of all diagonalized n-sided polygons with d diagonals $\mathcal{Q}_{n,d}$. The poset of all diagonalized n-sided polygons, ordered such that one diagonalized polygon P_1, satisfies $\mathrm{P}_1 \preceq \mathrm{P}_2$ if P_1 can be obtained from P_2 by adding diagonals, is denoted by $\mathcal{Q}_n$.

3.2 Defining Associahedra

We are now in position to give a definition of the central structure studied in this article.

Definition 11 (Associahedron)**.** The Associahedron of dimension n, denoted K_{n+2}, is the polytope whose face poset is isomorphic to $\mathcal{Q}_{n+3}$.

This means that all 0-cells (vertices) of K_n correspond to all fully diagonalized $(n+1)$-sided polygons, i.e. diagonalized polygons with $n-2$ diagonals. All 1-cells correspond to diagonalized polygons with $n-3$ diagonals, 2-cells correspond to diagonalized polygons with $n-4$ diagonals and so on.

What determines whether or not two vertices should be connected in K_n? From the definition of K_n it follows that two vertices are connected if the diagonalized polygons corresponding to the vertices differs from each other by only one diagonal. This can easily be realized when considering the Hasse diagram of $\mathcal{Q}_{n+1}$ which by definition is isomorphic to the face poset of K_n.

The more general question at this point could be: what determines which cells are contained in another in K_n? Again, by the definition of K_n, a k-cell is contained in a $(k+1)$-cell if and only if all diagonals in the diagonalized polygon corresponding to the $(k+1)$-cell are shared by the diagonalized polygon corresponding to the k-cell.

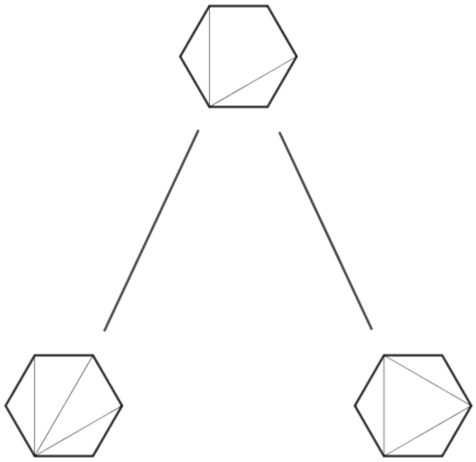
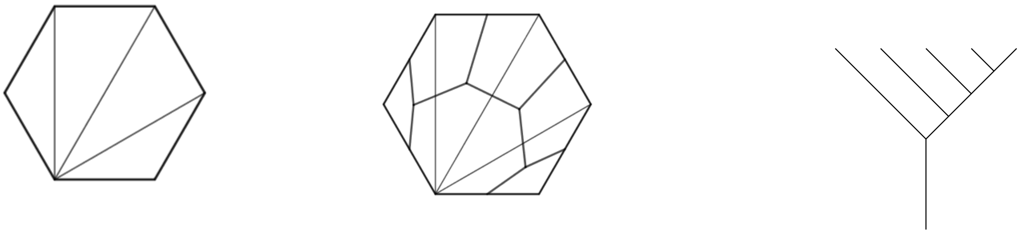


Figure 6: A part of the Hasse diagram of $\mathcal{Q}_6$, showing how a diagonalization is less than another diagonalization. The vertices of K_5 which correspond to the two polygons with three diagonals are connected with an edge in K_5 since they both are "less" than the diagonalized polygon with two diagonals corresponding to an edge.

3.3 The Correspondence Between PR-Trees and Diagonalized Polygons

There is a direct correspondence between PR-trees and the diagonalized polygons described above. Let $\mathcal{T}_n$ be the poset of all PR-trees with n leaves. Then $\mathcal{T}_n$ is isomorphic to $\mathcal{Q}_{n+1}$. This can be realized by letting every side in the polygon correspond to an outer vertex of the PR-tree and draw one inner vertex in every region of the diagonalized polygon and drawing one edge from that vertex to every side of that region. This procedure can be seen in Figure 7.

From this it follows that the associahedra K_n can also be defined as the polytope whose face poset is isomorphic to $\mathcal{T}_n$. In fact, the diagonalized polygons are mostly used to make it easier to understand the group action on the PR-trees. Before we proceed to the next section, we give one more definition. Let $\mathcal{T}_{n,m}$ be the set of all PR-trees which correspond to m-cells of K_n. $\mathcal{T}_{n,0}$ is then all PRB-trees corresponding to the vertices of K_n. As a consequence of what is mentioned above, $\mathcal{T}_{n,m}$ is isomorphic to $\mathcal{Q}_{n+1,n-2-m}$.



(a) Diagonalized polygon.

(b) In every region in 7a one places one vertex and draws an edge to all sides of that region.

(c) The tree which corresponds to the diagonalized polygon in Figure 7a.

Figure 7: How diagonalized polygons correspond to PR-trees.

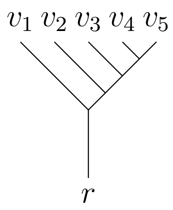
4 Group Actions on PR-Trees and Diagonalized Polygons

Consider the dihedral group of order n acting upon the diagonalized n-sided polygons. Clearly, this action can also be interpreted as an action on the PR-trees with $n+2$ leaves. All elements in D_n can be generated by rotations and reflections. The rotation partitions the set of all PR-trees into orbits in one way and the reflection in an other way.

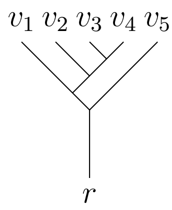
The rotation $\rho : \mathcal{T}_n \to \mathcal{T}_n$, easily understood on the polygons, $\rho($$) =$ , can be interpreted on the PR-trees as moving every outer vertex to the place of the outer vertex which is a neighbour to the original vertex in a counter-clockwise direction such that all connections between the vertices are preserved, i.e. consider the PRB-tree in Figure 8a, then a rotation as dictated by ρ maps $v_1 \mapsto r$, $v_2 \mapsto v_1$, $v_3 \mapsto v_2$, $v_4 \mapsto v_3$, $v_5 \mapsto v_4$ and $r \mapsto v_5$, yielding the tree in Figure 8b.

In appendix A one can find a table of PRB-trees, diagonalized polygons, orbits and more relevant information about dihedral group actions on PRB-trees with certain number of leaves. The orbits of the PRB-trees for K_5 is given below in table 1. In appendix B one can find a computer program that generates some of this information.

For K_5, there are 4 orbits of the vertices under the action of rotation of lengths 6, 3, 3 and 2, respectively. These orbits, the elements they correspond to, and the labels and



(a) PRB-tree with labeled vertices.

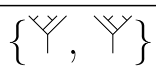
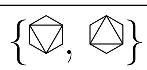
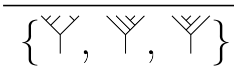
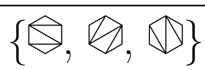
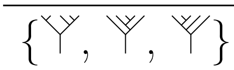
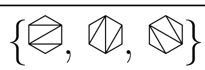
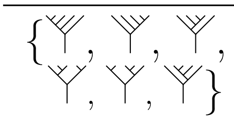
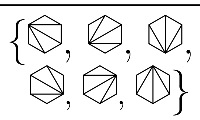


(b) Rotation of the tree in Figure 8a.

Figure 8: The tree to the right is a rotation of the tree to the left. The rotation can intuitively be understood either by transforming the tree to a diagonalized polygon, rotate the polygon and transfer back to a tree or by just considering the map $v_1 \mapsto r$, $v_2 \mapsto v_1$, $v_3 \mapsto v_2$, $v_4 \mapsto v_3$, $v_5 \mapsto v_4$ and $r \mapsto v_5$, which should leave all connections between the vertices preserved.

colours they are given in this paper can be seen in Table 1. In Table 2 one can find the orbits of the diagonalized polygons under the action of D_6 corresponding to edges of K_5. The same for faces can be found in Table 3.

Table 1: The orbits of the vertices of K_5 under the action of rotation, the elements they correspond to and the labels and colours they are given in this paper.

PRB-trees	**Fully diagonalized hexagons**	**Labels**	**Colour**
		$\{r_1, r_2\}$	red
		$\{y_1, y_2, y_3\}$	yellow
		$\{b_1, b_2, b_3\}$	blue
		$\{g_1, g_2, g_3,$ $g_4, g_5, g_6\}$	green

5 Symmetries of K_5

Before we proceed to investigating the symmetries of K_5, we define a *rigid motion*. Note that the definition of a rigid motion may be different in other articles.

Definition 12 (Rigid motion of $\mathbb{R}^n$)**.** A rigid motion of $\mathbb{R}^n$ is a composition of reflections and rotations, fixing the origin, of $\mathbb{R}^n$.

A rigid motion of $\mathbb{R}^n$ can be seen as rotations and reflections of an n-dimensional

Table 2: The orbits of the PR-trees under the action of rotation corresponding to the edges of K_5. In the left column one element of the orbit is visualized. All other elements in the orbit are rotations of the elements in the left column. The number of such elements can be found in the column in the middle. In the right column each orbit is labeled by a specific colour. Note that the colours are chosen arbitrarily and that there is no connection to the colours in table 1.

Diagonalized hexagon	**Rotations of the hexagon**	**Colour**
	6	
	6	
	6	
	3	

Table 3: The orbits of all diagonalizations of a hexagon consisting of one diagonal under the action of D_6. These correspond to faces of K_5.

Diagonalized hexagon	**Rotations of the hexagon**
	6
	3

sphere with center at the origin. A rigid motion of a polytope is a rigid motion of $\mathbb{R}^n$ such that all k-cells are mapped to another k-cell. We also define a rigid motion of the set of all k-cells of a polytope. This is a rigid motion of $\mathbb{R}^n$ such that every k-cell is mapped to another k-cell.

5.1 K_5 embedded in a Sphere

Since the associahedron is defined through PR-trees, the D_6 action on the PR-trees as described in section 4 must have a geometric interpretation. We propose that the group action on the PR-trees with n leaves corresponds to a rigid motion of at least one realization of K_n. This can easily be shown to be the case for K_4. We will now analyze the symmetries of K_5 and show that this is the case for K_5.

How different vertices should be connected in order to obtain K_5 was explained in section 3.2. What colour and label each vertex should get when considering dihedral group actions are described in table 1. Now, let these vertices be mapped to a sphere in such a way that the two red vertices are the north and south pole respectively. Place the blue and yellow vertices on the equator such that the blue and yellow vertices alternate and that the distance between consecutive vertices is the same. Three green vertices are then placed on the northern hemisphere in a symmetric way and the same is done for the remaining three green vertices on the southern hemisphere[2]. The result one should get is something like Figure 9.

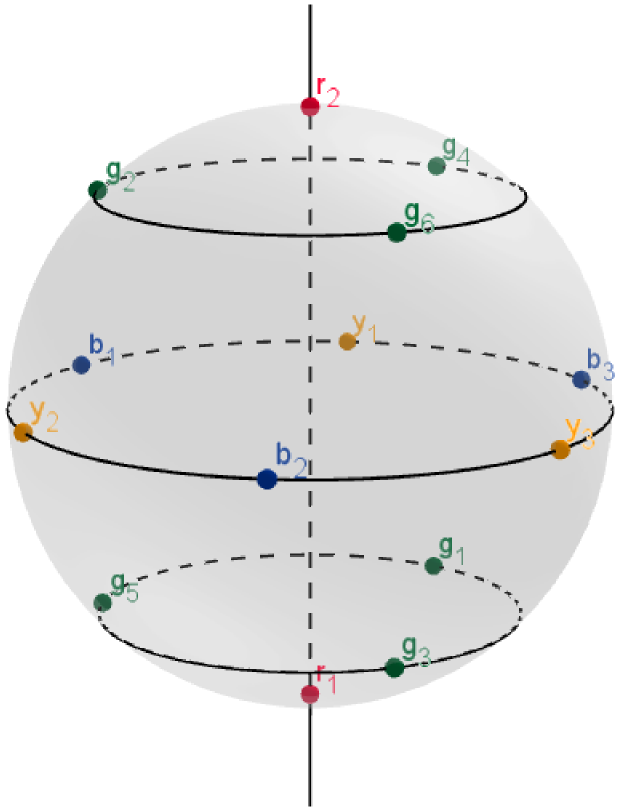


Figure 9: The coloured vertices placed on a sphere. The colours tell in which orbit the corresponding PRB-tree lies in when D_6 acts upon all PRB-trees with 5 leaves. Note that the lines in this picture do not represent the way vertices should be connected in order to obtain K_5 but is used to illustrate the symmetric way in which vertices are placed on the sphere.

From this point is it not hard to connect the vertices with edges in the correct way as described earlier. The result is K_5 since the combinatorial structure is inherent in the

[2]The PRB-trees corresponding to the green vertices forms an orbit when they are under the action of rotation. However, these "green" PRB-trees can also be separated in two different orbits if one considers the action of two rotations. Then these two orbits consist of three elements each, corresponding to the three elements that are positioned on the north and south pole respectively.

connections of the vertices. We have now obtained a symmetric version of K_5 embedded in a sphere. This can be seen in Figure 10. In Figure 11 the same realization is done but here the edges are coloured instead of the vertices according to Table 2.

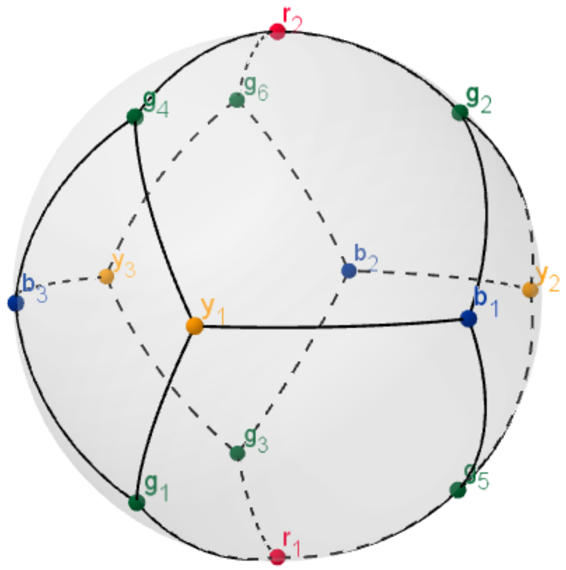


Figure 10: K_5 embedded in a sphere with vertices labeled according to which orbit their corresponding PRB-tree lies in under action of D_6.

5.2 Rigid Motions of K_5

We will now move on to investigate some rigid motions of K_5 as it is realized in this paper. Consider a rotation of $2\pi/3$ radians in anti-clockwise direction with respect to Figure 10 of K_5 which fixes r_1 and r_2. This is a rigid motion $\phi_{2\pi/3} : S \to S$ whose mapping can be seen on the vertices like:

$$\begin{aligned} &(\mathrm{r}_1, \mathrm{r}_2, \mathrm{y}_1, \mathrm{y}_2, \mathrm{y}_3, \mathrm{b}_1, \mathrm{b}_2, \mathrm{b}_3, \mathrm{g}_1, \mathrm{g}_2, \mathrm{g}_3, \mathrm{g}_4, \mathrm{g}_5, \mathrm{g}_6) \\ \mapsto\ &(\mathrm{r}_1, \mathrm{r}_2, \mathrm{y}_2, \mathrm{y}_3, \mathrm{y}_1, \mathrm{b}_2, \mathrm{b}_3, \mathrm{b}_1, \mathrm{g}_5, \mathrm{g}_6, \mathrm{g}_1, \mathrm{g}_2, \mathrm{g}_3, \mathrm{g}_4) \end{aligned} \tag{1}$$

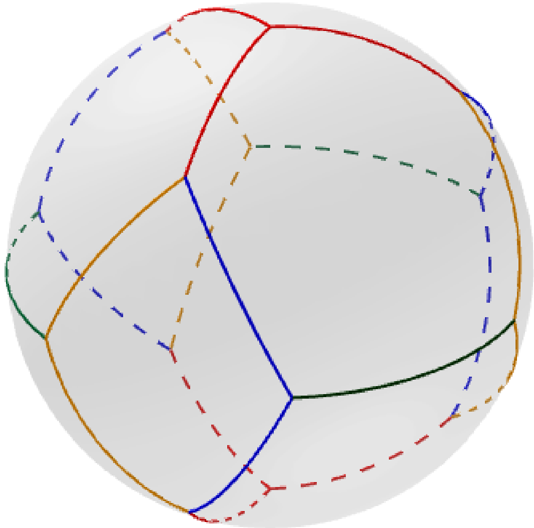


Figure 11: The edges of K_5 coloured as in Table 2.

The reflection through the equatorial plane forms a rigid motion $\Phi_{\mathrm{eq}} : S \to S$ which maps the vertices in the following way:

$$\begin{aligned} &(\mathrm{r}_1, \mathrm{r}_2, \mathrm{y}_1, \mathrm{y}_2, \mathrm{y}_3, \mathrm{b}_1, \mathrm{b}_2, \mathrm{b}_3, \mathrm{g}_1, \mathrm{g}_2, \mathrm{g}_3, \mathrm{g}_4, \mathrm{g}_5, \mathrm{g}_6) \\ \mapsto &(\mathrm{r}_2, \mathrm{r}_1, \mathrm{y}_1, \mathrm{y}_2, \mathrm{y}_3, \mathrm{b}_1, \mathrm{b}_2, \mathrm{b}_3, \mathrm{g}_4, \mathrm{g}_5, \mathrm{g}_6, \mathrm{g}_1, \mathrm{g}_2, \mathrm{g}_3) \end{aligned} \tag{2}$$

Lastly, consider the reflection through the plane crossing r_1, r_2, g_3 and g_6. This plane is perpendicular to the equatorial plane. The reflection is a rigid motion $\Phi_{\mathrm{orth}} : S \to S$ which maps the vertices according to:

$$\begin{aligned} &(\mathrm{r}_1, \mathrm{r}_2, \mathrm{y}_1, \mathrm{y}_2, \mathrm{y}_3, \mathrm{b}_1, \mathrm{b}_2, \mathrm{b}_3, \mathrm{g}_1, \mathrm{g}_2, \mathrm{g}_3, \mathrm{g}_4, \mathrm{g}_5, \mathrm{g}_6) \\ \mapsto &(\mathrm{r}_1, \mathrm{r}_2, \mathrm{b}_1, \mathrm{b}_3, \mathrm{b}_2, \mathrm{y}_1, \mathrm{y}_3, \mathrm{y}_2, \mathrm{g}_5, \mathrm{g}_4, \mathrm{g}_3, \mathrm{g}_2, \mathrm{g}_1, \mathrm{g}_6) \end{aligned} \tag{3}$$

5.3 Symmetries of K_5 as Described by Groups

We would like to show that the action of the dihedral group on the PR-trees could be interpreted as a rigid motion of some realization of the associahedron. In Lemma 1 we show that this is the same as just considering the dihedral group action on the PRB-trees and the way in which this affects the associahedron.

Lemma 1. *Let $C(K_n)$ be a realization of K_n as a convex hull of the set of vertices $V(K_n)$ interpreted as points in $C(K_n)$, then the action of D_{n+1} on $\mathcal{T}_n$ corresponds to a rigid motion of $C(K_n)$ if and only if the action of D_{n+1} on $\mathcal{T}_{n,0}$ corresponds to a rigid motion of $V(K_n)$.*

Proof. Let ρ be the action of rotation and ξ be the action of reflection. Then all elements of D_{n+1} can be generated by ρ and ξ and it is sufficient to investigate the action of these two elements. Let C_k be a k-cell of $C(K_n)$ and $D(C_k)$ the corresponding diagonalized polygon. From the definition of K_n, $D(C_k)$ can be seen as all diagonals shared by those $D(C_{k-1})$ such that $C_{k-1} \subset C_k$. This means that the order relation is preserved in the poset $\mathcal{Q}_{n+1}$ under the action of D_{n+1}, i.e. $\rho(D(C_{k-1})) \preceq \rho(D(C_k))$ and $\xi(D(C_{k-1})) \preceq \xi(D(C_k))$. Hence, all cells in K_n are connected in the same way as before the group action.

Since the position of a k-cell can be reduced to the positions of all vertices $V(K_n)$, a rigid motion of the vertices as dictated by the action of D_n must also be a rigid motion of all other cells if the rigid motion is defined separately for all k-cells. Since the combinatorial structure of K_n is preserved under the action of the dihedral group, Lemma 1 follows. $\square$

Lemma 1 says that the group action on the PR-trees leaves the ordering unchanged, and hence, D_{n+1} action on $\mathcal{T}_n$ leaves the combinatorial structure of K_n preserved. Furthermore, since the positions of all cells in a realization of K_n as a convex hull of a set of points is determined only by the vertices, the dihedral group action on the vertices determines the way other cells are effected by the group action.

Remark. Lemma 1 also applies to the realization of K_5 embedded in a sphere as described in section 5.1. This is true because the way different cells are built up in the realization are defined solely by the vertices.

We are now in position to present the main result of this article.

Theorem 1. *Let ρ be the action of rotation and ξ the action of reflection on $\mathcal{T}_5$ as dictated by the dihedral group. Let $C(K_5)$ be the realization of K_5 as described in section 5.1 and $V(K_5)$ the vertices in this realization. Let $\psi : \mathcal{T}_5 \to C(K_5)$ be a map which maps every PR-tree to the corresponding cell of K_5. The action of D_6 on $\mathcal{T}_5$ is then equivalent to rigid motions of $C(K_5)$, more precisely:*

$$\begin{tikzcd} \mathcal{T}_5 \arrow[r, "\psi"] \arrow[d, "\rho"'] & C(K_5) \arrow[d, "\Phi_{\mathrm{eq}}\circ\phi_{2\pi/3}"] \\ \mathcal{T}_5 \arrow[r, "\psi"] & C(K_5) \end{tikzcd} \qquad \begin{tikzcd} \mathcal{T}_5 \arrow[r, "\psi"] \arrow[d, "\xi"'] & C(K_5) \arrow[d, "\Phi_{\mathrm{orth}}\circ\Phi_{\mathrm{eq}}"] \\ \mathcal{T}_5 \arrow[r, "\psi"] & C(K_5) \end{tikzcd}$$

Proof. According to Lemma 1 the theorem follows if one can show that the rotation ρ and reflection ξ maps each element j in $\mathcal{Q}_{6,3}$ to another element i in $\mathcal{Q}_{6,3}$ such that the corresponding vertex to j in $C(K_5)$ is mapped by the rigid motion to the corresponding vertex of i. By function composition of the rigid motions defined in section 5.2 this can easily be checked to be true. The function composition of Φ_{eq} and $\phi_{2\pi/3}$ is seen on the vertices like:

$$\begin{aligned} &(\mathrm{r}_1, \mathrm{r}_2, \mathrm{y}_1, \mathrm{y}_2, \mathrm{y}_3, \mathrm{b}_1, \mathrm{b}_2, \mathrm{b}_3, \mathrm{g}_1, \mathrm{g}_2, \mathrm{g}_3, \mathrm{g}_4, \mathrm{g}_5, \mathrm{g}_6) \\ \mapsto\ &(\mathrm{r}_1, \mathrm{r}_2, \mathrm{y}_2, \mathrm{y}_3, \mathrm{y}_1, \mathrm{b}_2, \mathrm{b}_3, \mathrm{b}_1, \mathrm{g}_5, \mathrm{g}_6, \mathrm{g}_1, \mathrm{g}_2, \mathrm{g}_3, \mathrm{g}_4) \\ \mapsto\ &(\mathrm{r}_2, \mathrm{r}_1, \mathrm{y}_2, \mathrm{y}_3, \mathrm{y}_1, \mathrm{b}_2, \mathrm{b}_3, \mathrm{b}_1, \mathrm{g}_2, \mathrm{g}_3, \mathrm{g}_4, \mathrm{g}_5, \mathrm{g}_6, \mathrm{g}_1) \end{aligned}$$

Which is equivalent to the way ρ maps the corresponding PRB-trees or fully diagonalized polygons. This equivalence can be realized if one understands Table 1. Now, consider the function composition of Φ_{orth} and Φ_{eq}, this one is seen on the vertices like:

$$\begin{aligned} &(\mathrm{r}_1, \mathrm{r}_2, \mathrm{y}_1, \mathrm{y}_2, \mathrm{y}_3, \mathrm{b}_1, \mathrm{b}_2, \mathrm{b}_3, \mathrm{g}_1, \mathrm{g}_2, \mathrm{g}_3, \mathrm{g}_4, \mathrm{g}_5, \mathrm{g}_6) \\ \mapsto\ &(\mathrm{r}_1, \mathrm{r}_2, \mathrm{b}_1, \mathrm{b}_3, \mathrm{b}_2, \mathrm{y}_1, \mathrm{y}_3, \mathrm{y}_2, \mathrm{g}_5, \mathrm{g}_4, \mathrm{g}_3, \mathrm{g}_2, \mathrm{g}_1, \mathrm{g}_6) \\ \mapsto\ &(\mathrm{r}_2, \mathrm{r}_1, \mathrm{b}_1, \mathrm{b}_3, \mathrm{b}_2, \mathrm{y}_1, \mathrm{y}_3, \mathrm{y}_2, \mathrm{g}_2, \mathrm{g}_1, \mathrm{g}_6, \mathrm{g}_5, \mathrm{g}_4, \mathrm{g}_3) \end{aligned}$$

Which is equivalent to reflecting the PRB-trees or fully diagonalized polygons. $\square$

A question of interest at this point is what type of rigid motion of K_5 the action of different elements of D_6 correspond to. In Table 4 this information can be found. Note that all rigid motions can be explained by compositions of some other rigid motions. Furthermore, note that every rigid motion is a composition of three different rigid motions, each chosen from the three sets: $\{\Phi^0_{\text{orth}}, \Phi_{\text{orth}}\}$, $\{\Phi^0_{\text{eq}}, \Phi_{\text{eq}}\}$ and $\{\phi^0_{2\pi/3}, \phi_{2\pi/3}, \phi^2_{2\pi/3}\}$, where Φ^0_{orth}, Φ^0_{eq} and $\phi^0_{2\pi/3}$ correspond to not doing anything and therefore these are not spelled out. Since each element is chosen from all three "categories", there is a total of $2 \cdot 2 \cdot 3 = 12$ rigid motions corresponding to each of the twelve elements of D_6.

Table 4: In the left column the different elements of D_6 are listed. In the right column we show the effect this element have on K_5 when the element acts on $\mathcal{T}_5$, i.e. the poset which is isomorphic to the face poset of K_5.

Element in D_6	**Corresponding rigid motion of K_5**
e	No effect on K_5
ρ	$\Phi_{\text{eq}} \circ \phi_{2\pi/3}$
ρ^2	$\phi^2_{2\pi/3} = \phi_{4\pi/3}$
ρ^3	Φ_{eq}
ρ^4	$\phi_{2\pi/3}$
ρ^5	$\Phi_{\text{eq}} \circ \phi^2_{2\pi/3} = \Phi_{\text{eq}} \circ \phi_{4\pi/3}$
ξ	$\Phi_{\text{orth}} \circ \Phi_{\text{eq}}$
$\xi\rho$	$\Phi_{\text{orth}} \circ \phi_{2\pi/3}$
$\xi\rho^2$	$\Phi_{\text{orth}} \circ \Phi_{\text{eq}} \circ \phi_{4\pi/3}$
$\xi\rho^3$	Φ_{orth}
$\xi\rho^4$	$\Phi_{\text{orth}} \circ \Phi_{\text{eq}} \circ \phi_{2\pi/3}$
$\xi\rho^5$	$\Phi_{\text{orth}} \circ \phi_{4\pi/3}$

6 Further Research

There are several questions related to our study that might be interesting to address. Since we made a sphere embedding of K_5 which preserves the symmetries described by D_6, it is natural to ask if K_n can be embedded in a hypersphere in a similar manner. It would be interesting if the procedure in which we embedded K_5 in a sphere could be formalized and applied to K_n in such a way that Theorem 1 holds.

Another question of interest is if there exists a formula or easy procedure which can determine the lengths of all orbits of the PR(B)-trees under the action of rotation. If this can be found, then one can ask how this is related to the associahedra.

References

[1] Müller-Hoissen F, Marcel Pallo J, Stasheff J. *Associahedra, Tamari Lattices and Related Structures*. Basel: Birkhäuser; 2012.

[2] Stasheff J. (1963). *Homotopy Associativity of H-spaces.* In: *Transactions of the American Mathematical Society.*

[3] Lee, C. W. (1989). *The associahedron and triangulations of the n-gon. European Journal of Combinatorics*, 10(6), 551-560.

[4] Loday, J. L. (2005). *The multiple facets of the associahedron. preprint.*

[5] Devadoss, S. L., Forcey, S., Reisdorf, S., Showers, P. (2015). *Convex polytopes from nested posets. European Journal of Combinatorics*, 43, 229-248.

[6] Devadoss, S. L. (2009). *A realization of graph associahedra. Discrete Mathematics*, 309(1), 271-276.

[7] Benedetti B., Delucchi E., Moci L. (2015). *Combinatorial Methods in Topology and Algebra*. Paris: Springer.

A Orbits of Diagonalized Polygons and PR-Trees Under the Action of D_n

In Table 5 one can find information about the orbits of the PRB-trees under the action of rotation. The number of PRB-trees with n leaves is the $(n + 1)$:th Catalan number. The number of orbits is also an identifiable sequence [7].

Note that the lengths of the orbits always divides the order of the group which acts on the elements. This is known as the orbit-stabilizer theorem and is related to Lagrange's theorem which we described in section 2.1. This together with the fact that the rotation of a diagonalized polygon cannot be itself implies that whenever the number of sides of the polygon is a prime number, then all orbits will have that prime number as length.

Table 5: Orbits of PRB-trees under the action of rotation ρ and their relation to the associahedra.

n (dimension)	Number of PRB-trees	Lengths of the orbits	Number of orbits
1 (K_3)	2	$[2]$	1
2 (K_4)	5	$[5]$	1
3 (K_5)	14	$[2,3,3,6]$	4
4 (K_6)	42	$[\underbrace{7,...,7}_{6}]$	6
5 (K_7)	132	$[\underbrace{4,...,4}_{5},\underbrace{8,...,8}_{14}]$	19
6 (K_8)	429	$[3,3,\underbrace{9,...,9}_{47}]$	49
7 (K_9)	1430	$[\underbrace{5,...,5}_{14},\underbrace{10,...,10}_{136}]$	150
8 K_10	4862	$[\underbrace{11,...,11}_{442}]$	442

B Computer Program Generating the Orbits of $\mathcal{Q}_{n+3,n}$ Under the Action of Rotation

This program, written in Python 3.6.4rc1, calculates all possible diagonalized polygons and their orbits under the action of the dihedral group. For large[3] n this program is very slow. The program is split up into two modules.

```
def list_in_other(list1, list2):
    'returns True if list1 is in list2 e.g. [2, 3, 4] is in [1,
      2, 3, 4, 13, 0]'
    for i in range(0, len(list2) - len(list1)):
        for j in range(len(list1) - 1, len(list2)):
            if list2[i:j] == list1:
                return True
                break
            else:
                continue
    else:
        return False


def list_of_diagonals(length):
    'Return a list of all diagonals in an (length + 2)-gon'
    the_list = []
    basic_list = [x for x in range(0, length + 2)]
    for i in range(0, length + 1):
        for j in range(1, length + 3):
            element = basic_list[i:j]
```

[3]"large" is quite comparative.

```
            if j > i and len(element) <= length:
                the_list.append(element)
    the_list = [x for x in the_list if len(x) != 1]
    return the_list


class PolygonDiagonalization:

    def __init__(self, diagonals):
        self.diagonals = [sorted(x) for x in diagonals] #The
            class input should be a list of diagonals
        self.size = len(self.diagonals) + 3
        self.vertices = [x for x in range(0, self.size)]
        edges = []
        for vertex in self.vertices:
            if [vertex] + [vertex + 1 % self.size] not in edges
                or [vertex + 1 % self.size] + [vertex] not in
                edges:
                edges.append([vertex] + [vertex + 1]) #Makes
                    list of edges
        self.edges = edges

    def diagonal_equality(self, d1, d2):
        'Returns True if two diagonals in a polygon are equal'
        if sorted(d1) == sorted(d2):
            return True
        elif sorted(d1 + d2) == self.vertices:
            return True
        else:
```

```python
            return False

    def diagonal_replacer_1(self, d1):
        'For a diagonal in a polygon, this function returns the
            other representation of the diagonal'
        d2 = sorted([x for x in self.vertices if x not in d1])
        return d2

    def diagonal_replacer_2(self, d1):
        'For any diagonal in a polygon, this function returns a
            specific representation'
        d2 = self.diagonal_replacer_1(d1)
        if len(d1) == len(d2):
            if min(d1) < min(d2):
                return d1
            else:
                return d2
        elif len(d1) < len(d2):
            return d1
        else:
            return d2

    def __repr__(self):
        return '@' + str(self.diagonals) + '!'

    def __eq__(self, other):
        self_diagonals = [self.diagonal_replacer_2(x) for x in
            self.diagonals]
```

```
        other_diagonals = [other.diagonal_replacer_2(x) for x in
            other.diagonals]
        if sorted(self_diagonals) == sorted(other_diagonals):
            return True
        else:
            return False

    def __ne__(self, other):
        self_diagonals = [self.diagonal_replacer_2(x) for x in
            self.diagonals]
        other_diagonals = [other.diagonal_replacer_2(x) for x in
            other.diagonals]
        if sorted(self_diagonals) == sorted(other_diagonals):
            return False
        else:
            return True

    def rotation(self):
        'Generates a new object corresponding to the rotations
            of self'
        new_diagonals = []
        for diagonal in self.diagonals:
            new_diagonal = []
            for x in diagonal:
                new_diagonal.append((x + 1) % self.size)
            new_diagonals.append(sorted(new_diagonal))
        return PolygonDiagonalization(new_diagonals)
```

```python
def diagonal_in_other(self, d1, d2):
    'Returns True if d2 encloses d1'
    if list_in_other(d1, d2)\
       or list_in_other(d1, self.diagonal_replacer_1(d2))\
       or list_in_other(self.diagonal_replacer_1(d1), self.
          diagonal_replacer_1(d2))\
       or list_in_other(self.diagonal_replacer_1(d1), d2):
        return True
    else:
        return False


def diagonal_intersect(self, d1, d2):
    'Returns True if d1 and d2 intersect each other'
    d1 = self.diagonal_replacer_2(d1)
    d2 = self.diagonal_replacer_2(d2)
    cut = [diagonal for diagonal in d1 if diagonal in d2]
    if cut != [] and cut != d1 and cut != d2:
        return True
    elif self.diagonal_equality(d1, d2) == True:
        return True
    else:
        return False


def diagonal_switch(self, d1):
    'Returns a new object with the diagonal d1 changed'
    diagonals_1 = sorted([diagonal for diagonal in self.
       diagonals if not self.diagonal_equality(diagonal, d1)
       ])
```

```
        list_ = list_of_diagonals(self.size - 2)
        for diagonal_1 in list_:
            for diagonal_2 in diagonals_1:
                if self.diagonal_intersect(diagonal_1,
                   diagonal_2):
                    break
            else:
                if not self.diagonal_equality(d1, diagonal_1):
                    diagonals_1.append(diagonal_1)
                    break
        return PolygonDiagonalization(diagonals_1)

    def rotation_cycle(self):
        'Generates a list of all rotations of self'
        a = self
        list_ = []
        while list_.count(a) < 1:
            list_.append(a)
            a = a.rotation()
        for a in list_:
            for b in list_:
                if list_.index(a) != list_.index(b):
                    if a == b:
                        list_.remove(b)
        return list_

    def diagonal_switches(self):
```

```
        'Generates a list of objects corresponding to all new
           diagonalizations created by changing diagonals'
        list_ = []
        for diagonal in self.diagonals:
            list_.append(self.diagonal_switch(diagonal))
        for a in list_:
            for b in list_:
                if list_.index(a) != list_.index(b):
                    if a == b:
                        list_.remove(b)
        return list_

    def rotational_equality(self, other):
        'Returns True if other is a rotation of self'
        rot = other.rotation_cycle()
        if self in rot:
            return True
        else:
            return False
```

```
from class_polygon_diagonalization import *

n = int(input('Dimension : '))

a = PolygonDiagonalization([range(0, m) for m in range(2, n + 2)
   ])

all_ = [element for element in a.rotation_cycle()] #List of all
```

```
    diagonalizations
orbits = [a.rotation_cycle()] #List of all orbits
orbits_0 = [element for element in orbits]
k = 0


while k == 0:
    new_cycle_length = [] #Determines when to stop the loop
    for cycle in orbits_0: #Looking at each orbit separately
        last_in_cycle = cycle[len(cycle) - 1] #Defining the last
            element in the orbit
        new_cycle = last_in_cycle.diagonal_switches() #Creating
           a list of all elements \
        #which can be obtained from the last element of the
           orbit by flipping one diagonal (list = new_cycle)
        for cycle_0 in orbits: #Removing each element from
           new_cycle which already is in orbits
            for element in cycle_0:
                if element in new_cycle:
                    new_cycle.remove(element)
        #Removing duplicates in new_cycle
        new_cycle_copy = [element for element in new_cycle]
        for element_1 in new_cycle_copy:
            for element_2 in new_cycle_copy:
                if element_1 != element_2 and element_1.
                   rotational_equality(element_2):
                    if element_2 and element_1 in new_cycle:
                        new_cycle.remove(element_2)
        #End of duplication-remove
```

```
        if len(new_cycle) == 0: #Determines when to stop the
            loop
            new_cycle_length.append(0)
        else:
            new_cycle_length.append(1)
        #Appending:
        for element in new_cycle: #All element which are not in
            all_ or orbits
            rot = element.rotation_cycle() #Cycle of all
                rotations for all new diagonal configurations
            orbits.append(rot) #Appending new orbit to orbits
            for element_0 in rot: #Appending new element to all_
                all_.append(element_0)
    if 1 not in new_cycle_length: #Determines when to stop the
        loop
        k = 1
    orbits_0 = [element for element in orbits]

#Printing out the data
print()
print('The number of fully diagonalized polygons with %d sides :
    ' % (n + 3))
print(len(all_))
print()
orbits_lengths = sorted([len(element) for element in orbits])
print('Orbit lengths : ')
for element in orbits_lengths:
    print(element, end = ', ')
```

```
print()
print()
print('The number of orbits : ')
print(len(orbits))
```

C Exact Coordinates for the Sphere-embedding of K_5

In Table 6 one can find the exact coordinates of the vertices of K_5 embedded in a sphere as explained in section 5.1.

Table 6: Spherical coordinates for the realization of K_5 as described in section 5.1.

Vertex	**Spherical coordinates**
r_1	$(1, 0, -\frac{\pi}{2})$
r_2	$(1, 0, \frac{\pi}{2})$
y_1	$(1, \frac{\pi}{6}, 0)$
y_2	$(1, \frac{5\pi}{6}, 0)$
y_3	$(1, \frac{3\pi}{2}, 0)$
b_1	$(1, \frac{\pi}{2}, 0)$
b_2	$(1, \frac{7\pi}{6}, 0)$
b_3	$(1, \frac{11\pi}{6}, 0)$
g_1	$(1, 0, \frac{\pi}{4})$
g_2	$(1, \frac{2\pi}{3}, \frac{\pi}{4})$
g_3	$(1, \frac{4\pi}{3}, -\frac{\pi}{4})$
g_4	$(1, 0, \frac{\pi}{4})$
g_5	$(1, \frac{2\pi}{3}, -\frac{\pi}{4})$
g_6	$(1, \frac{4\pi}{3}, \frac{\pi}{4})$