UNIVERSITÉ LIBRE DE BRUXELLES

Faculté des Science

Département d'Informatique

Rolling Regular-faced Polyhedrons on k-Uniform Tessellations: Rolling properties and Reachable Space

V1.3 J12 Hyperlink fix

Akira BAES

Promotor : Prof. Stefan LANGERMAN - Master Thesis in Computer Sciences

Academic year 2020 - 2021

Acknowledgements

I would like to thank

Prof. Sterfan LANGERMAN for showing me his research for related subjects, sharing his code, telling me that the question was interesting enough to write about and encouraging me to continue doing research by sharing his experience and networking, as well as for the hours spent revising my text. I was glad to have the opportunity to work on this subject.

Elizabeth HARTUNG and Aaron WILLIAMS from the Barbados research workshop for showing interest and passion as researchers, testing my code, giving feedback and sharing useful ideas and directions, which helped me get out of an unproductive slump.

My parents for raising me with an interest in mathematics, art and videogames, for allowing me to continue my studies for so long, and for giving me an environment in which I could completely focus on studying.

My brother and grandmother for always being supportive.

Lastly, the various people on the internet who came to discuss puzzles and make my research feel important.

Abstract

In this work we observe regular-faced polyhedron *rolling* on grids made of periodic regular-faced tessellations in the plane. A plane roller is a shape that can roll in the whole space for a given tiling. We describe a method to prove the belonging to the classification. We discuss the properties of various rollers for use in puzzles.

Contents

Part I

Main thesis body

Chapter 1

Introduction

Humans have used tessellation patterns, polyhedrons, and the interactions between shapes and underlying grids since ancient times. Square grids in particular were the theater for board games like chess with a grid explored by pieces moving in a particular pattern. Cubic dices being common has lead to a form of puzzles, rolling block puzzles, where a cube dice rolls on a square grid with a pattern imposed by its faces. The popularity of such puzzles would grow with the advent of computer games, which made playing them more convenient. And now in this work, other polyhedron and tessellation rolling pairs are being explored.

1.1 Literature review

We will review the history of the different concepts that will come together in this work.

1.1.1 Tessellations

Historically, the foundation for the classification of regular tilings starts in 1619 with Kepler's Book II of *Harmonices Mundi* [1] listing the 11 now called Archimedeans or 1-uniform tilings of the plane. Many authors started building arbitrary classification methods. Matila Ghyka coined the concept of semi-regular tiling in 1946[2]. D. Chavey would continue exploring it in 1984[3]. The definition of demiregular tilings would be finalised in 2006 by H. Aslaken et al. [4] The term generalised parallelogon for tilings using translations would appear in B. Grünbaum and G. C. Shephard's *Tiling with congruent tiles* in 1980[5]. Isogonal tilings were considered in 1916 by A. V. Subnikov and by others since then. B. Grünbaum and G. C. Shephard would list the 91 types of isogonal tilings in the plane in 1978 in an eponymous work[6], and the 81 isohedral tilings in 1977[7]. They would create a system for k-uniform tilings by regular polygons in 1977[8], and D. Chavey would continue their catalog in 1989[9]. Grünbaum and Shepard published a book called *Tilings and patterns* regrouping many of the modern discussion and findings in 1987[10]. P. Hofman makes his own catalog of tilings to include radial aperiodic tilings in 2019[11]. Tilings research continues as representations of hyperspace tilings are observed in Hyperbolic grids and discrete random graphs by E. Kopczyński and D. Celińska-Kopczyńska in 2017[12].

1.1.2 Polyhedrons

Work on classification of all regular-faced polyhedron includes Euclid's proof of the five Platonic solids[13], convex regular polyhedra. Archimedes' lost proof, and Kepler rediscovering and classifying the 13 Archimedean solids and the prisms and antiprisms[14]. The rest of the regular-faced polyhedrons, that would end up called the Johnson solids, were described in *The Faces of a Regular-Faced Polyhedron* 1965 by B. Grünbaum and N. W. Johnson[15], and in *Convex Polyhedra with Regular Faces* 1966 by Norman W. Johnson[16], and finally proved in *Convex polyhedra with regular faces* by V. A. Zalgaller in 1969[17].

1.1.3 Rolling

Rolling a polyhedron to cover the space in some form has been explored by *Rolling Polyhedra on a Plane, Analysis of the Reachable Set* by Y. Chitour et. al. in 1997[18], *Reachability and Steering of Rolling Polyhedra* by A. Bicchi et al. in 2004.[19] Rolling a polyhedron to stamp the plane in *Tile-Makers and Semi-Tile-Makers* J. Akiyama 2007[20]. Unrolling a polyhedron's net as a tessellation in 2011 *Determination of all Tessellation Polyhedra with Regular Polygonal Faces* by J. Akiyama et. al.[21]. Tessellation Polyhedron would be the first polyhedron and tiling pairs that were researched in the context of this thesis, before branching out to a more general expression of tilings.

It can be noted that research on polyhedron rolling on tilings accelerated when it has become easier to manipulate such polygonal objects in puzzle games virtually.

1.1.4 Rolling cube puzzles

A lot of rolling cube puzzles descriptions date from the mathematical games columns of Martin Gardner in *Scientific American* that began in 1957. They are explored in works such as *Rolling*

Block Mazes are PSPACE-complete, K. Bunching, M. Bunchin, 2012[22], *On the Complexity of Rolling Block and Alice Mazes* by M. Holzer, S. Jakobi in 2012.[23], *Minimum moves of Rolling cube puzzles*, J. Yao 2021[24]. In particular, *On Rolling Cube Puzzles* by K. Bunchin, M. Bunchin, et. al. in 2007[25] mentions the possibility of using different Polyhedron/Tiling pairs than the classic Cube and square grid.

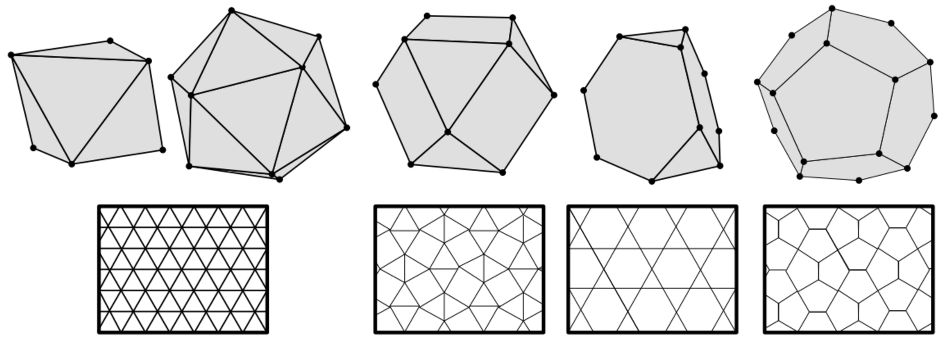


Figure 5: Solids with suitable grids. From left to right: octahedron, icosahedron, cuboctahedron, truncated tetrahedron, dodecahedron.

Figure 1.1: Tiling/polyhedron pairs from ”On Rolling Cube Puzzles”

1.1.5 Rolling puzzles in Videogames

Videogames taking advantage of the tile-polyhedron compatibility of the cube and the square grid are numerous. Here are some examples.

- *Edge.* iOS, 2008
- *Kyub.* Windows, 2016
- *Cube* from *Simon Tatham's Portable Puzzle Collection.* Web, 2021
- *The Last Cube.* Windows, 2022

One puzzle type in particular, where you have to match the face of a cube to a tile on the plane, has been a practical mechanic in many videogames. Here is a list of examples:

- *Korodice.* Gameboy, 1990 (dice face matching)

- *Devil Dice.* Playstation, 1998 (neighbour face matching)
- *Legacy of Kain: Soul reaver.* Playstation, 1999 (pipes matching)
- *Legend of Zelda Oracle of Ages.* Gameboy Color, 2001 (colored face matching)
- *Bombastic.* Playstation 2, 2002 (neighbour face matching)
- *Legend of Zelda Spirit Tracks*, Nintendo DS, 2009 (colored face matching)
- *Rubek.* Windows, 2016 (paint colored face matching)
- *Roll The Box.* Mobile, 2021 (indicated face matching)
- *Pip Push Paradise.* Windows, 2021 (dice face matching)

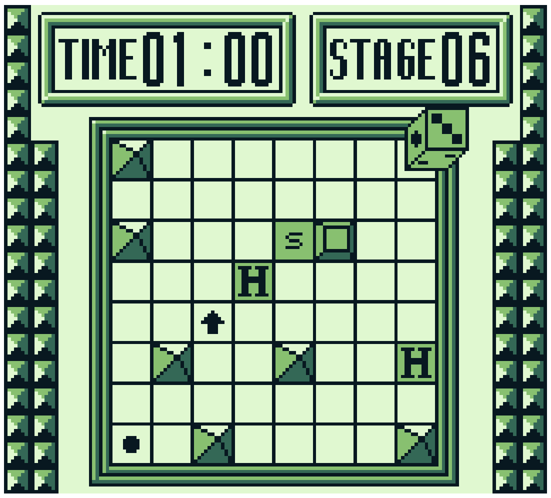
Like shown, entire games revolving around the mechanic of rolling are still being made, and the genre might benefit from the results described in this work.

Non-square grid games

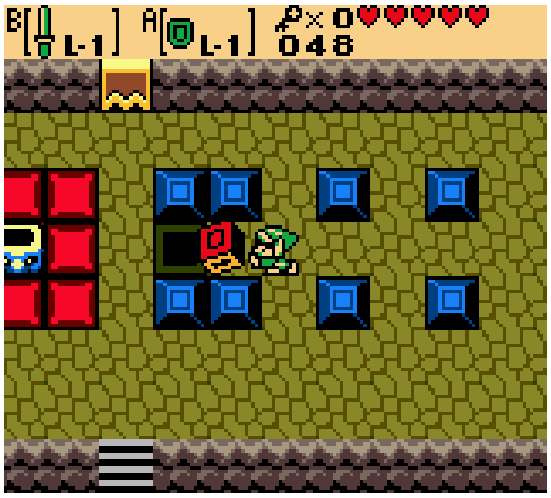
Games that use different tiling than the square grid exist, mostly boardgames, mostly hexagonal grids, and rarely, pentagonal.
One notable videogame is *HyperRogue* (2015), which uses Hexagonal and Heptagonal tiles in a hyperspace. It also features face-matching rolling puzzles that do not feature a cube, instead featuring many other triangle-faced Platonic solids commonly used for dice, and roll them on a triangular lattice over its hexagonal grid.

1.2 Motivation

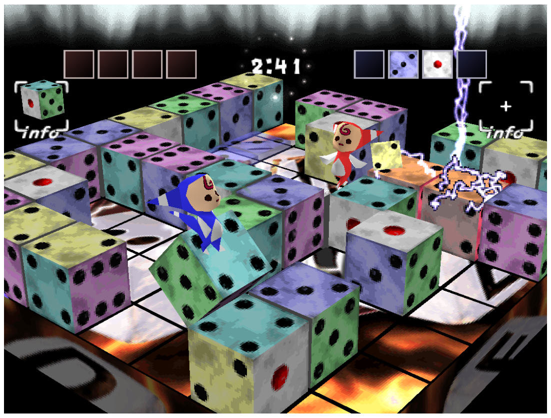
A general motivation is that, as shown previously, most rolling shape puzzles use cubes, and rolling a cube on the square grid is a very well researched and documented subject. However, the subject of rolling other shapes is currently poorly explored, limited mainly to platonic polyhedron on regular-adjacent tilings. As a puzzle games enthusiast, researching this subject feels as it could lead to greater developments in the domain, or at least inspire new puzzlemakers and puzzle types.
A personal motivation to research this subject was coming in contact with tessellation polyhedron, whose nets can tile the plane, imagining them roll in their nets, and wonder what patterns might emerge from those movements. As mathematical research sometimes does, the exploration of rolling patterns on various tilings can produce artistically valuable, visualy striking images. Being able to



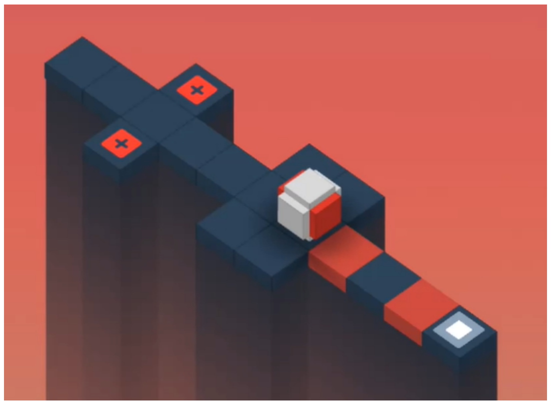
(a) *Korodice* (1990)
(captured on emulator)



(b) *Zelda Oracle of Ages* (2001)
(via https://www.zeldadungeon.net/)



(c) *Devil Dice* (1998)
(via emuparadise by user thebobble)



(d) *Rubek* (2016)
(capture from steam store page video)

Figure 1.2: Face-matching rolling cube puzzles

explore this new space in the context of a thesis, especially after struggling to find a subject, is a rare opportunity.

1.3 Scope of the work

The scope of this thesis is to define what a roller is, define an algorithm to determine if a regular-faced polyhedron and periodic tiling pairing form a roller, build tools to manipulate, observe rolling pairs and derive other properties from the rolling that would be useful when used in a game. We will also concretely observe all regular-faced convex polyhedrons on a limited number of tilings, and tessellation polyhedrons on one of their tilings.

1.4 Structure overview

We will start this thesis by defining various terms and concepts that will be used, separated by category.
We will then write a proof that can be turned into an algorithm to prove that a tiling/polyhedron pair is a plane roller. We will also define some additional properties and methods to check them.
We will go over aggregated results. The more in-depth results are available at the end of the work.
It is recommended to go back-and-forth between the imaged tables and the discussion.
We will highlight results that might be of interest in the context of the motivation: use in puzzle games.
We will conclude on what was discussed so far.
In the annex we will then dive further in the implementation of some parts and then I will list the software development process. We will end with a review of the rolling data collected in tables.

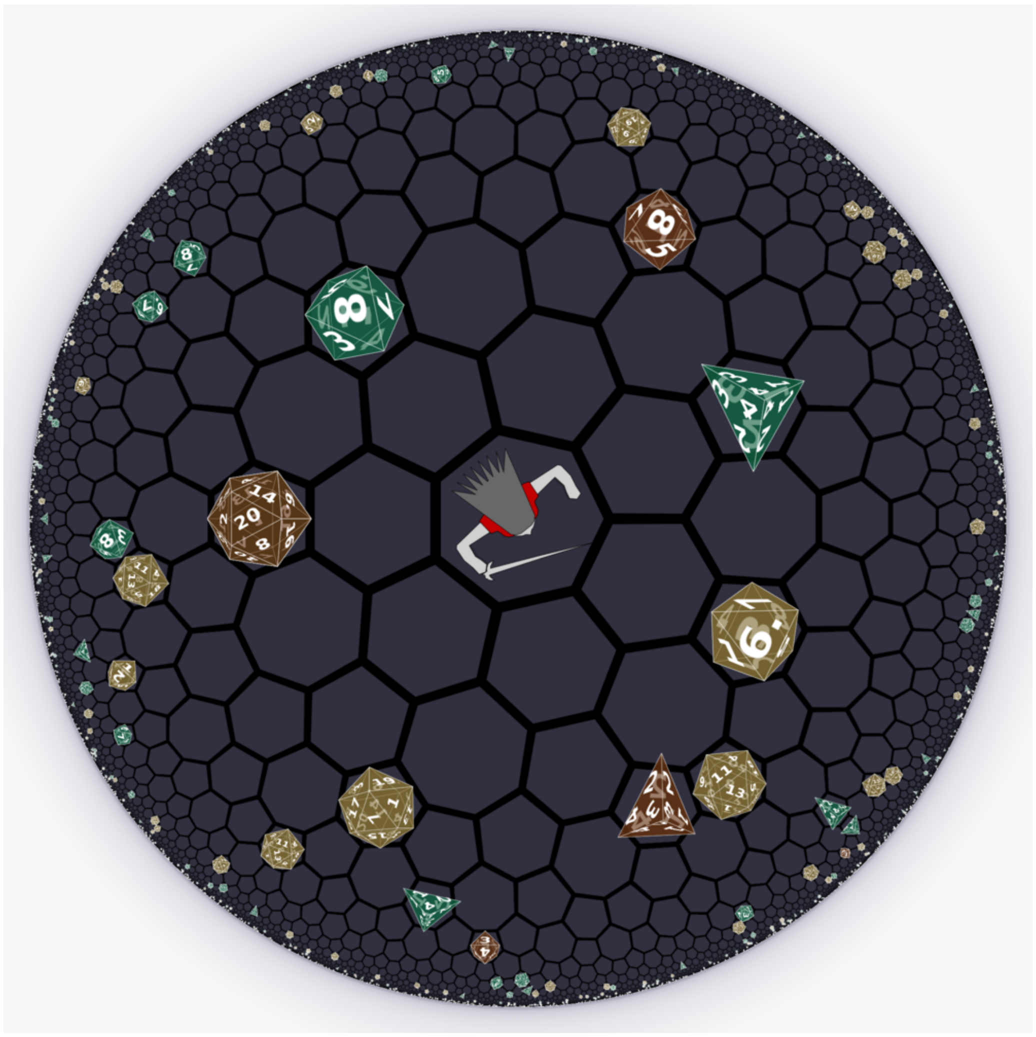


Figure 1.3: *HyperRogue* (2015) Dice Reserve update (2021)
(capture from steam store page)

Chapter 2

Definitions

We will define terms for tessellations, polyhedrons, the relations between the two and the concept of roller that derives from it, plus some properties.

2.1 Tessellations

Tessellations will serve as our grid for rolling. Some terms will be necessary to continue.

Definition 1 (tiling/tessellation/tiles)**.** A *tiling* or *tessellation* is a collection of plane figures in $\mathbb{R}^2$ or *tiles* that cover the plane with no gap or overlap [10].

Definition 2 (vertex/dual)**.** A tiling where polygons are aligned edge-to-edge describes a primary graph where the points at the intersection of tiles are the graph's vertices, and the edges between two tiles are the graph's edges. A vertex type is determined by the faces that surround it.
A tiling's dual graph, which will be our default representation of a tiling graph, takes the polygonal tiles faces as vertices, and the edges separating two faces as edges between those two faces. A face type is determined by the polygon of that face.

Definition 3 (prototiles)**.** A *prototile* is one tile which is repeated trough symmetries to form a tessellation. A tessellation can be formed of several prototiles. [26]

Definition 4 (monohedral, n-hedral tiling)**.** A *monohedral* tiling has every tile congruent (identical in size and shape) to the others trough a combination of translation, rotation and reflection. It has one unique prototile. A n-hedral tiling has n tile shapes.

Definition 5 (k-isohedral tiling)**.** A monohedral plane tiling is called *isohedral* if one symmetry group is enough to transitively describe all of its tiles. A *k-isohedral* tiling has k separate classes (or orbits) of tiles to be transitively symmetrical trough k finite sets of symmetries. The tiles can be differently shaped (n-hedral) or have the same shape (monohedral). [7]

Definition 6 (regular-faced tessellation)**.** We will call *regular-faced* any tessellation using exclusively one or more types of regular n-gons as prototiles. In this work we will only use edge-to-edge aligned regular-faced tessellation (all sharing the same edge length).

Definition 7 (k-isogonal tiling)**.** A tiling with one vertex type is called *isogonal* if one symmetry group acts transitively on its vertices. If it contains k vertex types that each are transitively symmetrical (trough k orbits), it is called k-isogonal.

Lemma 1. A k-isohedral tiling is at most k-isogonal.[9]

Definition 8 (k-uniform tiling)**.** A k-isogonal tiling only using regular polyhedron as prototiles is called k-uniform.

Definition 9 (periodic tiling)**.** A periodic tiling is a tiling that has a repeating translating pattern. k-isohedral or k-isogonal tilings with finite k are periodic as the symmetry classes form a repeating pattern that turns into a translation when looking at more tiles.

Definition 10 (Platonic tiling/Regular tiling)**.** The three regular-faced 1-isohedral tilings of the plane using one regular prototile with edge-to-edge transformations are the *Platonic tilings* or *Regular tilings*. They are the triangular grid (regular triangular tiling), the square grid (square tiling), and the hexagonal grid (regular hexagon tiling).

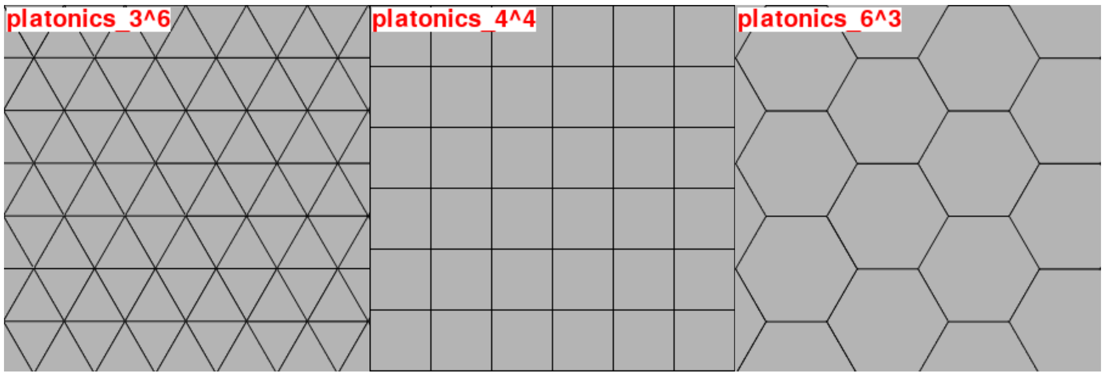


Figure 2.1: Platonic tilings

Definition 11 (Archimedean, 1-uniform, semiregular)**.** The 1-uniform tilings that are not 1-isogonal are called *Archimedean* or *semiregular* tilings.[9]

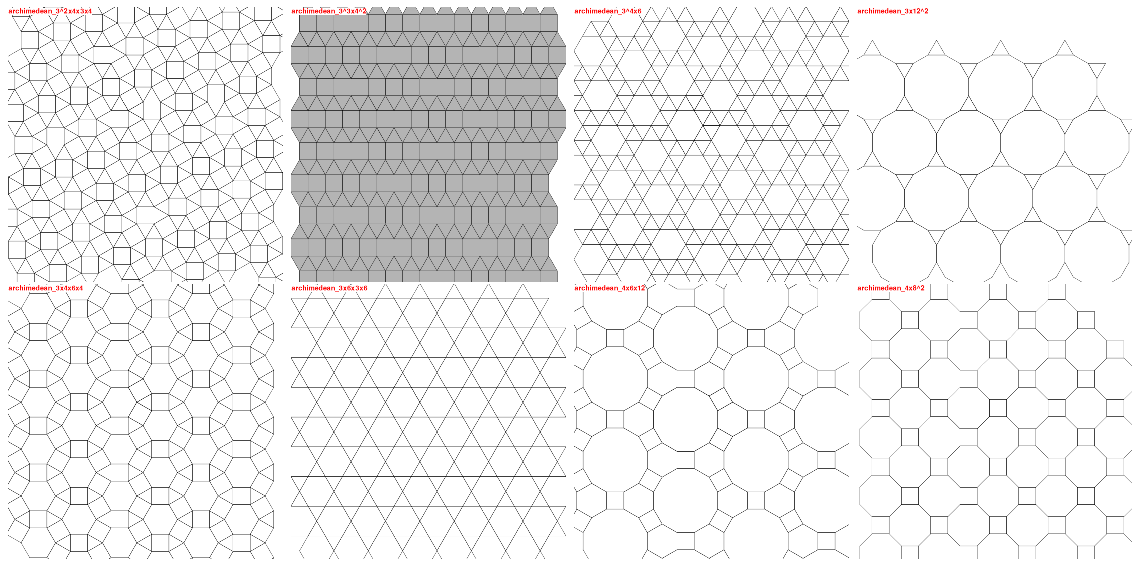


Figure 2.2: The 8 archimedean tilings

Note. In the following work we will use k-isohedral regular-faced tessellations that have a conway supertile tessellation description, in the order as they appear in the catalog [9] under their "Platonic, Archimedean, k-isogonal" classification paired with their respective vertex-orbit notation.

Definition 12 (tiling naming convention)**.** K-uniform tilings with increasing k can be classified. The catalogue[9] that was used for this work defines a naming convention for each tiling that we will also be using. It consists of an arrangement by increasing number of unique vertex types in the primary tiling graph. A vertex type is defined by a clockwise or counter-clockwise list of the number of sides of the polygonal faces that meet the vertex in the tiling, ordered from the smallest to the biggest with repeating n-gons compacted. A tiling is then defined by a list of vertex types separated by semicolons. If two tilings share their vertex type representation, they are differentiated by a subscript. For example, $(3^3 * 4^2; 4^4)_1$ and $(3^3 * 4^2; 4^4)_2$.
It is not necessary to imagine the tiling based on its name only, as a picture or a link will usually be provided.

Definition 13 (conway tessellation)**.** A *conway tessellation* is a 1-isohedral tessellation for which the prototile tiles the plane trough a translation and up to four 180° rotations. Each one corresponds to a neighbouring tile of the prototile. [27]

Definition 14 (Parallelogon)**.** We will call a *parallelogon* tile (generalised parallelogon) a prototile for a monohedral 1-isohedral tiling that tiles the space using only one class of translations (two translation axis).[5] It can also be called a Beauquier-Nivat tile.[28] There are parallelogram parallelogons for where each tile has four neighbours, and hexagonal parallelogon that have six.

Definition 15 (supertile and grid-tiling)**.** We define a monohedral tessellation whose prototile is a shape made of several polygons stitched together edge-to-edge. We call this tile the *supertile*. We call the tessellation described by the polygons forming the supertile its corresponding *grid-tiling*. The sub-tiles in the supertile are the prototiles of the grid-tiling, and they all share the same transformation classes. If a supertile is formed of regular polygons, its grid-tiling is a regular-faced tessellation.

Definition 16 (cell of supertile)**.** When talking about a polygonal sub-tile in relation to its supertile, we might refer to it as a *cell* of the supertile, identified by an index. When talking about the same sub-tile in relation to the plane in the grid-tiling, we will simply refer to it as a tile.

Definition 17 (conway supertile)**.** A*conway supertile* is a supertiling whose prototile is also a conway tile.

Definition 18 (parallelogon supertile)**.** A*parallelogon supertile* is a supertiling whose prototile is also a generalised parallelogon.

Lemma 2. For every periodic tessellation, there is a parallelogon supertile covering the repeated surface.

Definition 19 (Cartesian and skewed cartesian coordinates)**.** In a plane parallelogon tiles system, cartesian coordinates define each tile uniquely by a pair of coordinates along two translation unit axes. When the two axes are not perpendicular (for hexagonal parallelogons), the system is called axial or skewed coordinates system [29].

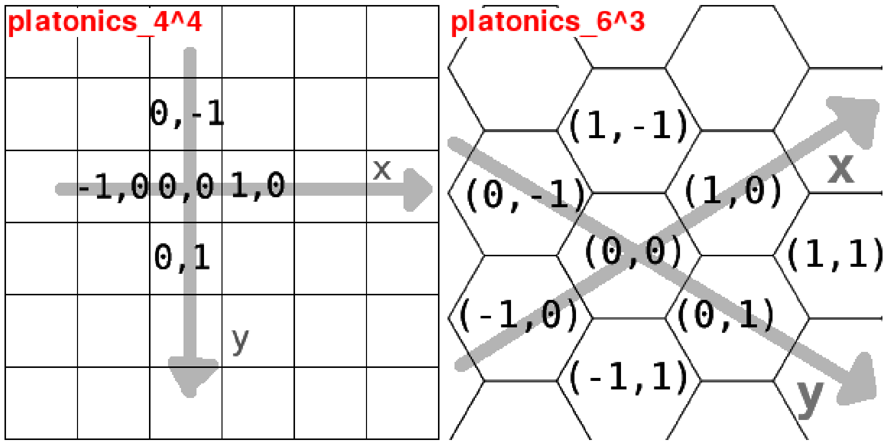


Figure 2.3: Regular parallelogon tilings with coordinates

Note. In the following work we will exclusively discuss periodic regular-faced tilings that we can describe trough repetitions of a base supertile. We will mostly use parallelogon supertiles which are easier to handle trough cartesian coordinates.

2.2 Polyhedrons

The polyhedron we will be rolling on the tiling.

Definition 20 (regular-faced polyhedron)**.** A *regular-faced polyhedron*is a convex polyhedron whose faces are regular polygon. This includes the five Platonic solids, the 13 Archimedean solids, prisms and antiprisms, and the 92 Johnson solids.

Note. In the following work we will use regular-faced polyhedron paired with various regular-faced tilings. For this reason we will only consider prisms and antiprisms sided 3,4,5,6,8,10,12, as those are the polygons that appear in tilings.

Definition 21 (net)**.** *A net* of a polyhedron is a non-overlapping spanning tree of the dual graph (faces) of the polyhedron arranged edge-to-edge.

Definition 22 (tessellation polyhedron)**.** A *Tessellation polyhedron* as defined in [21] is a polyhedron which possesses at least one net that can be used to isohedrally tile the plane. Such a net is called a *tessellation net* and the tiling of the plane is called a *net tessellation*.
A tessellation net is a *supertile* for the tiling described its faces.

Note. In [21] it was determined that out of all regular-faced polyhedrons, only 23 are tessellation polyhedron.

2.3 Tile and position

Once we have a tessellation and a polyhedron, we can define their relationship.
For the next definitions, let T be a tessellation in the plane (regular-faced, k-isogonal).
For the next definitions, let P be a regular-faced polyhedron that will roll into the tessellation.

Definition 23 (tiling graph, adjacent tiles)**.** Let t be a tile in the tessellation T. We note $t \in T$.
We explore the *tiling graph*, which is the dual graph (tiles) of the tessellation. We use the notation $N_i(t)$ for the i^{nd} adjacent tile of f, in cyclic counter clockwise order.

Definition 24 (faces graph, neighbouring faces)**.** Let f be a face of the polyhedron P. We note $f \in P$.
We explore the polyhedron *faces graph*, which is the dual graph (faces) of the polyhedron. We use $N_i(f)$ for the i^{nd} adjacent face of f, in cyclic counter clockwise order. —P— denotes the number of faces on the polyhedron.

Definition 25 (chirality)**.** The chirality of a shape is wether its points and sides are defined counter-clockwise or clockwise. We will only work with counter-clockwise chirality, even for shapes that are the result of a mirror transformation.

Definition 26 (face size)**.** We use the notation $|f|$, $|t|$ for the number of sides of a polygonal face/tile respectively. In this work, we consider regular-faced tessellation and regular-faced polyhedrons. Meaning that if $|f| = |t|$, both share the same regular n-gon shape.

Definition 27 (compatible tile)**.** Let f be a face of a regular-faced polyhedron P. Let t be a tile in a regular-faced tessellation T. If $|f| = |t|$, the face f shares the same shape the tile t, and f and t are *compatible*.

Definition 28 (orientation)**.** For every polygonal face f of a polyhedron P, we number each of the sides of its polygon counter-clockwise starting from one side that will serve as the reference side. This defines a static reference system called *orientation*. We do the same for every polygonal tile t of tessellation T, with one side being the reference orientation, and the next orientations being numbered in clockwise cyclical order.

Note. Since we describe plane tilings trough transoformations of a supertile, each tile's orientations should be based on its corresponding prototile cell's orientation.

Definition 29 (relative orientation)**.** A *relative orientation* between a polyhedron's face overlapping with a compatible tile is the orientation of the side of the face that overlaps the side of the tile of the reference orientation. This means that knowing the tile, face, and relative orientation, we know the exact position and rotation of the polyhedron on the tiling. From now on, the term orientation will refer to the relative orientation, and the tile orientation will be considered static.

Definition 30 (face sitting on a tile, state, position)**.** For a face f of a polyhedron P and a compatible tile $t \in T$, we say the polyhedron P *sits on the tile* t *in the tessellation* T *with its face* f *and (relative) orientation* o if f and t completely overlap and their relative orientation is o.
We call a $< f, o >$ tuple a *state*.
We call a $< t, f, o >$ tuple containing a state and a tile a rolling *position*.

2.4 Roller

Rolling is the specific relationship between a polyhedron and a tiling that we are interested in.

Definition 31 (roll, valid roll)**.** A roll is when a polyhedron P in a tessellation T changes its sitting face f to an adjacent face, and its sitting tile t to an adjacent tile in the same direction r (expressed as an orientation on t).
A roll in the direction r of P from a position $< t, f, o >$ to an adjacent position $< t', f', o' >$ is valid

if and only if

- The tile and face rolling in that direction correspond to the adjacent one in that direction:

$$< t, f, o > \xrightarrow{r} < N_r(t), N_{o+r}(f), o' > \tag{2.1}$$

- And the new orientation o' matches the reciprocal restriction:

$$o' \quad \text{s.t.} \quad N_{o'+r'}(N_{o+r}(f)) = f \tag{2.2}$$

  where r' is the index of t among the neighbours of $N_r(t)$, i.e. $N_{r'}(N_r(t)) = t$

Physically, rolling a convex polyhedron that is sitting on the plane is rotating the polyhedron about one of the edges of the sitting face until an adjacent face rests on the plane in a compatible tile. [18]

Definition 32 (rolling graph, roller)**.** A *rolling graph* for a polygon P on a tessellation T is a graph whose vertices are all possible $< t, f, o >$ positions where $f \in P$, $t \in T$ and where two vertices are connected if there exist a valid roll r between the two positions. A roller or rolling pair refers to a polyhedron and tiling pairing for which we explore the rolling graph.

Definition 33 (reachability, path)**.** *Reachability* is defined on a rolling graph for two positions u and v if there exists a *path* of valid rolls between u and v in the rolling graph (if the components are connected). The *reachable positions* from u is the set of positions that are connected to u. Notation: $v \sim u$: v is reachable from u.

Definition 34 (reachable area)**.** The *reachable area* from a position is the union of the tiles present in the reachable positions of a starting position. We note for a starting position u:

$$RA(u) = t \in T : \exists f, o \quad \text{s.t.} \quad u \sim < t, f, o > \tag{2.3}$$

Definition 35 (connected component)**.** Two sub-graph of a rolling graph are *connected* if they can reach each other's positions. A *connected component* of the rolling graph, is the sum of all the connected positions that can reach eachother.

Definition 36 (path)**.** A *path* between two reachable positions is the list of the direction of the intermediate valid roll positions used to connect the two positions. Because the graph is undirected, a path can be reversed by inverting it.

Definition 37 (plane roller)**.** A polyhedron P is a *plane roller* on a tiling T starting at position $< t_0, f_0, o_0 >$ iff every tile on the plane is present in the reachable area of $< t_0, f_0, o_0 >$ or

$$\forall t \in T \quad \exists f, o :< t, f, o > \sim < t_0, f_0, o_0 > \tag{2.4}$$

Depending on the context, the term roller is used interchangeably with plane roller.

Definition 38 (partial roller)**.** A roller for P on T at $< t_0, f_0, o_0 >$ is a *partial roller* if some faces are unused in the rolling graph, or $\exists x \in P : \nexists t, o :< t, x, o >\sim< t_0, f_0, o_0 >$. If all faces are used, it is a perfect roller.

Definition 39 (face-complete tile)**.** A roller for P on T at $< t_0, f_0, o_0 >$ has a *face-complete tile* iff all compatible faces of the polyhedron can roll on that tile with any orientation, or given $t \in T, \forall f \quad \text{s.t.} \quad |t| = |f| : \exists o \quad \text{s.t.} \quad < t, f, o >\sim< t_0, f_0, o_0 >$.

Definition 40 (face-complete roller)**.** A roller for P on T at $< t_0, f_0, o_0 >$ is a *face-complete roller* iff all tiles in the reachable area are face-complete.

Note. Face-complete rollers are frequently used in rolling puzzles. Face-complete rollers that exclude incompatible faces are partial rollers. The cube on the regular square grid tiling is a face-complete roller.

Definition 41 (face-orientation-complete tile)**.** A face-orientation-complete tile, or just orientation-complete tile is a tile that can be visited with all faces with every orientations in a connected part.

Definition 42 (orientation-complete roller)**.** A roller for P on T at $< t_0, f_0, o_0 >$ is a *face-orientation-complete roller* (or just orientation-complete) iff all compatible positions are reachable with every orientation, or $\forall < t, f, o > \quad \text{s.t.} \quad |t| = |f| :< t, f, o >\sim< t_0, f_0, o_0 >$.

Definition 43 (face-static roller)**.** A roller for P on T at $< t_0, f_0, o_0 >$ is a *face-static roller* iff for every tile $t \in G$ there exists only one f,o pair so that $< t, f, o >\sim< t_0, f_0, o_0 >$.
The tetrahedron on the regular triangular grid tiling is a staticroller. Static rollers usually are not used in face-matching puzzles as they only accept the trivial solution.

Note. The property of how many reachable (face, orientation) positions a (Polyhedron, Tiling) rolling pair has, is important in rolling shape puzzles, as the goal is often to match a given face/orientation on the polyhedron and the plane.[25] The closest to an orientation-complete roller it is, the more there are options and paths to create an interesting (non-trivial) puzzle.
For example the cube, which is used in many rolling puzzles, is face-complete on the square tiling.

2.5 Reachable area

Even if a polyhedron P is not a plane roller for a tiling T for a given $< t, f, o >$, we can observe the reachable area's shape. The definitions are provided here, the proof later via symmetry vectors in section 3.3.

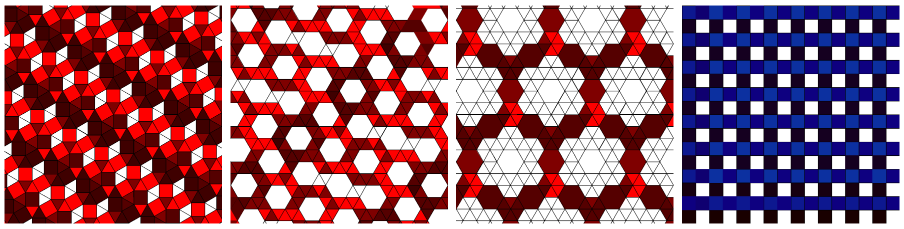


Figure 2.4: Hollow reachable space :
J35 in $(3^2 \times 4 \times 3 \times 4)$ — J87 in $(3^4 \times 6)$ — J55 in $(3^6; 3^4 \times 6)_1$ — J19 in (4^4)

Definition 44 (hollow plane roller)**.** The reachable area is a *hollow plane* if the reachable area extends infinitely in all directions but has holes.

Definition 45 (band roller)**.** The reachable area is a *band* if the reachable area extends infinitely in two opposite directions.

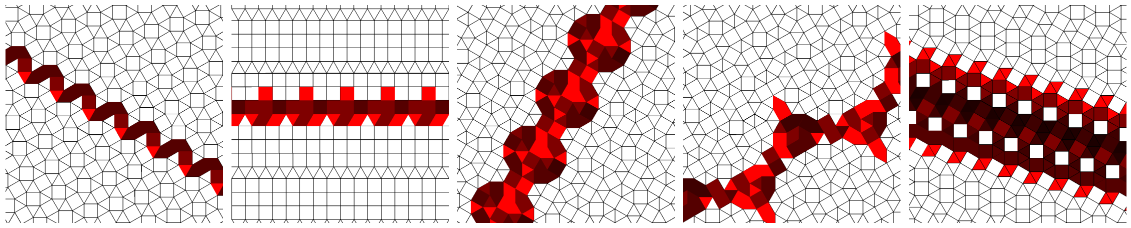


Figure 2.5: Reachable band :
J14 in $(3^2 \times 4 \times 3 \times 4)$ — J49 in $(3^3 \times 4^2; 4^4)_2$ — J26 in $(3^3 \times 4^2; 3^2 \times 4 \times 3 \times 4)_1$
J36 in $(3^3 \times 4^2; 3^2 \times 4 \times 3 \times 4)_1$ — snub cube in $(3^3 \times 4^2; 3^2 \times 4 \times 3 \times 4)_2$

Definition 46 (bounded roller)**.** The reachable area is *bounded* if the reachable area is limited and finite.

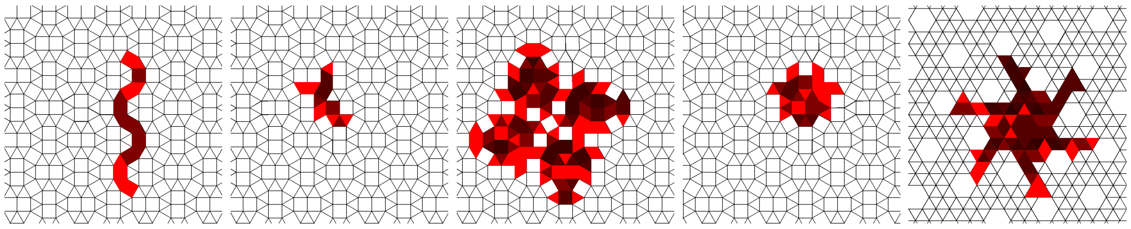


Figure 2.6: Bounded reachable space :
J74, J88, J44, J22 in $(3^3 \times 4^2; 3^2 \times 4 \times 3 \times 4)_1$ — J22 in $(3^6; 3^4 6)_2$

Definition 47 (stable tile)**.** A tile in the reachable area is *stable* if changing the face or the orientation of the polyhedron on the tile does not change its reachable area's shape.

Definition 48 (stable area)**.** A reachable area is stable if all its tiles are stable.

Definition 49 (stable roller)**.** A roller which has the whole plane as reachable area is stable if all the tiles in the plane are stable.

Definition 50 (quasi-plane roller)**.** A quasi-plane roller or quasi-roller has a reachable area that is a hollow plane, but the unreached areas are only tiles that are incompatible with the faces of its polyhedron. A quasi-roller can be stable if its reachable area is stable.

Note. Knowing stable tiles of a rolling pair is an important property to know where to put the starting position for a rolling graph and be guaranteed to reach the defined rolling area without having to account for face and orientation details.

Chapter 3

Proving the rolling property

Outline: Based on definition 37 a *plane roller* has a connected component in its rolling graph that contains every tile in the tiling. Connectivity can be proven by exploring the graph. If we can find a small connected component and extend it trough symmetries to prove that it can cover the whole space, we can prove a plane roller. We will first prove the rolling property on square and hexagonal regular tilings to take advantage of the easy to define parallelogon coordinates system. Then we will turn every rolling described on a regular-faced periodic tiling into a rolling on a parallelogon tiling by aggregating states by supertiles instead.

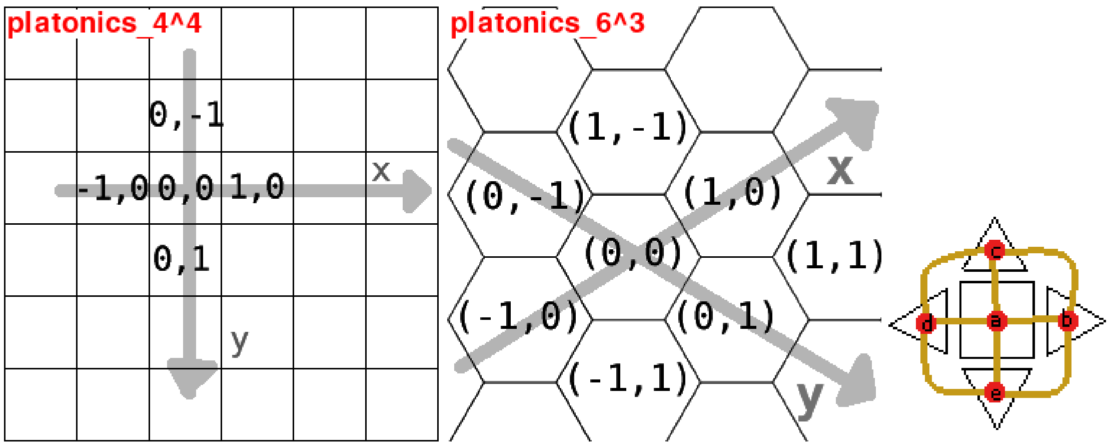


Figure 3.1: Regular tilings with parallelogon coordinates, and a polyhedron dual graph

3.1 Rolling on a regular parallelogon tiling

To simplify the expression of translating tile coordinates and rolling path, let's first observe the most basic case of rolling on a regular tiling (all same tiles) that uses a parallelogon (translated squares, hexagons) with an easy to define coordinates system.
Let $< f, o > \in P$ be rolling states of the polyhedron
Let $t \in T$ be a tiling of periodicity of one tile on cartesian coordinates (x,y) as described in fig. 3.1
Let $< t_0 = (x_0, y_0), f_0, o_0 >$ be the starting $< t, f, o >$ position in the rolling graph.

Fact 1. For regular grid tilings, the number of unique possible $< f, o >$ rolling graph states for each tile is

$$N_{regular} = \sum_{f \in P} |f| \quad \text{if} \quad |f| == |t| \qquad \forall t \in T \tag{3.1}$$

Fact 2. For every rolling path of size $N_{regular}$ on the rolling graph, there is a repeated $< f, o >$ state, by pigeonhole principle.

Fact 3. From the starting position we can reach a tile with every possible and reachable $< f, o >$ state in $N_{regular}$ steps.

Proof. if
$\exists \quad path =< f_0, o_0 >\sim< f_1, o_1 >\sim ... \sim< f_N, o_N > \quad | \quad N \geq N_{regular}$
then by Fact 2:
$\exists < f_i, o_i >, < f_j, o_j > \in path \quad | \quad i < j \quad and \quad < f_i, o_i > = < f_j, o_j >$
and $< f_N, o_N >$ is reachable by a shorter path
$< f_0, o_0 >\sim ... \sim< f_i, o_i > + < f_j + 1, o_j + 1 >\sim ... \sim< f_N, o_N >$
of size $N - (j - i)$ until $N <= N_{regular}$ □

Two tiles reached in the same connected portion with the same $< f, o >$ state inform of a periodic translation symmetry class in the structure of the rolling graph.

Lemma 3.

$$\begin{gathered}\text{if} < (x, y), f, o >\sim< (x + u, y + v), f, o > \quad \text{defines } u, v \text{ a symmetry} \\ \text{then} \quad \forall < (x', y'), f', o' >\sim< (x, y), f, o >: \\ < (x', y'), f', o' >\sim< (x' + k * u, y' + k * v), f', o' > \qquad \forall k \in \mathbb{Z}\end{gathered} \tag{3.2}$$

Proof. Let $< (x, y), f, o >\sim< (x', y'), f, o >$ two symmetrical positions where $(x', y') = (x+u, y+v)$ and the path between the two.
Because the tiling structure repeats, all the positions of the path can be translated by u, v to create

a path between $< (x', y'), f, o >$ and $< (x' + u, y' + v), f, o >$
The path can be reversed to create a path between $< (x', y'), f, o >$ and $< (x' - u, y' - v), f, o >$
By transitivity the states symmetry class described applies to every state in the rolling graph that is connected to the starting state.

□

Definition 51 (base)**.** We call a set of linearly independent symmetry vectors that can generate all the other symmetries a *base*.

We want to be able to create a base by looking at a limited number of tiles.

Lemma 4. If a base exists, there exists a minimal base where each vector is generated by a path of length equal or shorter than $N_{regular}$

Proof. By Fact 3 we know that for every $< t, f, o >$ we can find another position with the same state trough a path of length $\leq N_{regular}$. Let's consider all of those positions to build a set of vectors W.
We must at least find enough different positions in W that define linearly independent vectors to build a base.
Let's pretend that W is not enough to build a base. This means that there is a position X with the state $< f, o >$ that requires a path longer than $N_{regular}$ to reach, but cannot be reached by a combination of vectors in W.
By Fact 2 this path has a structure $< t_0, (f, o) > \sim < t_1, (f, o) > \sim < t_X, (f, o) >$ with a repeating $< f, o >$ state inserted at a distance $\leq N_{regular}$ (repeat as many times as necessary). This means that the vector of X a linear combination of $< t_0, (f, o) > \sim < t_1, (f, o) >$ which should be in W and $< t_1, (f, o) > \sim < t_X, (f, o) >$, which by symmetry should also exist in W. This is a contradiction so X does not exist and W is enough to build a base. □

This lead to:

Corollary 1. We can define a base for the symmetries of a rolling graph by rolling in an area of size $N_{regular}$ around the starting position.

Theorem 1. We can recreate the whole rolling graph by exploring all paths of size $N_{regular}$ around the starting position and translate them with the symmetry vectors.

Proof. Trough Corollary 1 we can find all symmetry vectors in paths of size $\leq N_{regular}$
Trough Lemma 3 we can reach every tiles that shares a $< f, o >$ trough translations
Trough Fact 2 we reach every $< f, o >$ variations in paths of size $\leq N_{regular}$ □

Let's formalise that last part:

Definition 52. We define T_s a minimal set of reachable tiles with unique $< f, o >$ states with path length$\leq N$ (Fact 3). $T_s \subseteq T_r$

$$\forall < f, o > \exists 1! < t, f, o > \in T_s \tag{3.3}$$

For convenience, let's define a contiguous set of tiles to use:

Definition 53. The representative area T_r is the area of the tiling of size described by a parallelogram formed by the symmetry vectors around the starting tile that can be used to recreate the whole rolling graph.

Theorem 2. If the base for the symmetries of a rolling graph contains only one linearly independent element, the reachable area is a band formed by translating the representative area along the axis. If the base for the symmetries of a rolling graph contains no element, the reachable area is bounded by the representative area.

Theorem 3. A plane roller and hollow-plane roller has two linearly independent elements in its symmetries base.

Theorem 4. A polyhedron P on a regular tiling T starting on $< t_0, f_0, o_0 >$ is a plane roller iff
$\forall t \in T_r : \exists < t_s, f_s, o_s > | < t_s, f_s, o_s > \sim < t_0, f_0, o_0 >$
and $\quad < t_s, f_s, o_s > \in T_s$
and $\quad t$ can be reached trough symmetries of ts

In practice, this means exploring the representative area T_r for all unique connected $< f, o >$ states, either directly or by wrapping a path trough translation symmetry. This is enough to prove that the whole plane can be reached, because the wrapping can explore arbitrarily long paths while the total number of states to explore remains finite in the parallelogram.
If the parallelogram testing area has holes after that operation, we are looking at a hollow-plane roller.

3.2 Rolling on a semi-regular tiling

Geometrically we can apply the previous logic to regular-faced periodic tilings as well, by using the cartesian distance in $\mathbb{R}^2$ between the centers of tiles on the plane to define the translation vectors for the base. But computationally, using real space coordinates as a way to build a base for tiles that are connected by different angles is prone to rounding errors. For the sake of the implementation, we will define another cartesian integer coordinates system.

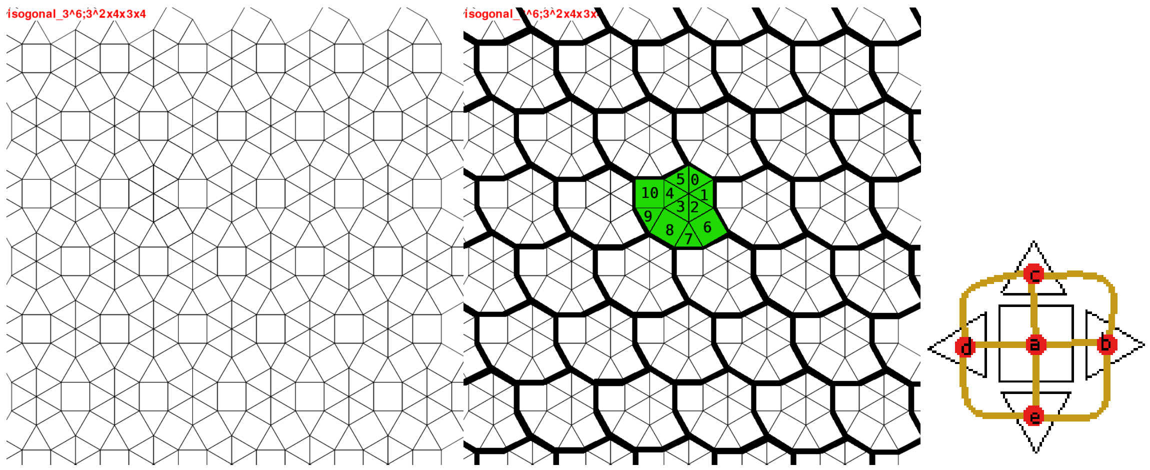


Figure 3.2: Tiling T ; Supertile tiling ST with cells in green ; Polyhedron P

3.2.1 Semi-regular tiling rolling graph

For a semi-regular, periodic tiling, we cannot define the same $< f, o >$ states per tiles, because not all tiles allow for all $< f, o >$. We take advantage of the period in the tiling to redefine a few element:

- T a k-isohedral (periodic) **regular-faced grid-tiling** formed of regular **polygonal tiles**

- S a related parallelogon **supertile** formed of $|S|$ *tiles* with **cells index** describing the periodic symmetries of the grid-tiling

- $c_t \in S$ a corresponding **cell index** of the tile t on the *prototile supertile* S

- ST the plane tiling made of supertiles
- $t \in T$ has a corresponding $t \in S$

Combining the previous variables, we now temporarily have redefined our system:

- a position $< t, c_t, f, o >$ in the rolling graph with
- a corresponding $< c_t, f, o >$ state that defines the periodic rolling graph.

Fact 4. The number of possible $< c_t, f, o >$ periodic rolling graph states is:

$$N_{periodic} = \sum_{c_t \in S_0} \sum_{f \in P} |f| \quad \text{if} \quad |f| = |t| \qquad \forall S \in ST \tag{3.4}$$

Fact 5. For every rolling path of size $N_{periodic}$ on the rolling graph, there is a repeated corresponding $< c_t, f, o >$ state by pigeonhole principle.

We could define symmetries for different types of tiles by comparing the central point of tiles and implement the same proofs, but it is harder to write symmetry class proofs that use tile paths because of the mixed shapes. To facilitates this while having an integer symmetry vectors, we define a new rolling graph on the aggregated rolling states over the supertiles:

3.2.2 Supertile rolling graph

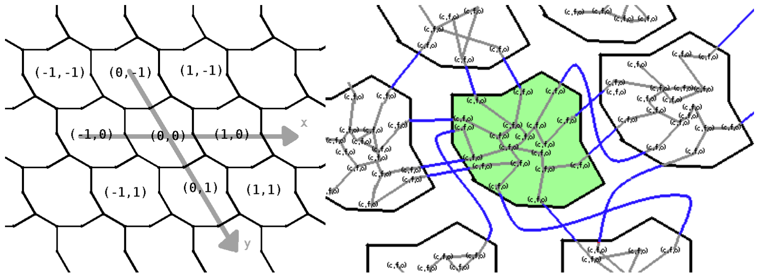


Figure 3.3: Left: supertile parallelogon tessellation with a coordinates system.
Right: one super-position (green) made of one connected part of the internal rolling graph of P over the supertile S, linked to other super-positions trough external rolling graph (blue).

Definition 54. A *supertile rolling graph* is composed of:

- $S \in ST$ **S**upertiles in the **S**upertile **T**iling
- sp, or **S**uper**P**osition an aggregation of $< c, f, o >$ positions that are connected inside a supertile without leaving it (internal rolling graph)
- $b(S, sp) = (< S', sp' >, < S'', sp'' >, ...)$ the aggregated set of valid rolls at the border or a supertile with a super-position. We use this to roll form one super-position to another, from one supertile to another, like we would with a polyhedron and tiling pair.

Fact 6. For supertile rolling graph, the number of possible *sp* states (N_{sp}) is the number of connected portions in a supertile. It is smaller or equal than the corresponding periodic rolling graph, depending on the size of the supertile: $N_{sp} <= N_{periodic}$. If the supertile is made of one regular sub-tile, it is equal to $N_{regular}$.

Fact 7. For every rolling path of size N_{sp} in the supertile rolling graph, there is a repeated superstate sp by pigeonhole principle.

Because we defined earlier supertiles as parallelogons, it is easy to define paths, symmetries similarly as for regular parallelogon tilings, with a cartesian coordinate system. We substitute:

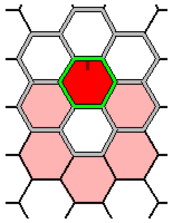
- $t \in T \longrightarrow S \in TS$
- Coordinates of t $\longrightarrow$ Coordinates of S
- Dual graph of P $\longrightarrow$ Super-positions graph sp,b
- State $< f, o > \longrightarrow$ super-position sp
- $< t_0 = (x_0, y_0), f_0, o_0 > \longrightarrow < S_0 = (x_0, y_0), sp_0 > \mid < f_0, o_0 > \in sp_0, t_0 \in S_0$
- $N_{regular} \longrightarrow N_{sp}$

Doing this, all the other properties for symmetry vectors and representative area apply.

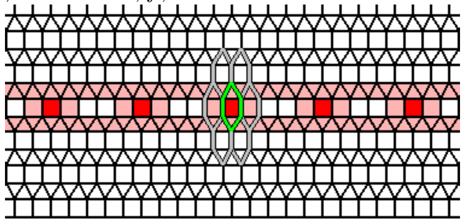
Theorem 5. A polyhedron P on a regular-faced tiling T starting on $< t_0, f_0, o_0 >$ is a roller iff its corresponding SuperTiling ST, superpositions sp over b_{st}, starting in sp_0 that contains $< t_0, f_0, o_0 >$ is a roller iff
$\forall t \in T_r : \exists < t_s, f_s, o_s >, sp_s | < t_s, f_s, o_s > \in sp_s \quad and \quad sp_s \sim sp_0$
and $\quad sp_s \in T_s$
and $\quad t$ can be reached trough symmetries of t_s

3.3 Additional properties

Theorem 6. If the rolling graph does not have symmetry vectors, the reachable area is bounded and P,T over $< t, f, o >$ is not a roller.



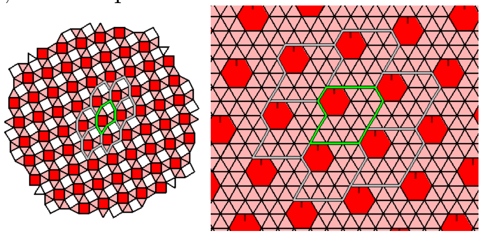
Theorem 7. If the rolling graph only has one linearly independent vector, the reachable area is a band and P,T over $< t, f, o >$ is not a roller.



Theorem 8. If the rolling graph has two linearly independent vectors, the reachable area extends infinitely in all directions.

If not every tile t is present in the reachable supertiles, the reachable tiles is a hollow plane and P,T over $< t, f, o >$ is not a roller, but could be a quasi-plane roller if the tiles that could not be reached do not have a compatible face in P.

If every tile t is present in the reachable supertiles, the reachable tiles is a plane and P,T over $< t, f, o >$ is a plane roller.



Theorem 9. A plane roller pair P,T is stable on a tile $t \in T$ if

$$\forall f \in P \quad \text{s.t.} \quad |f| == |t|, \quad \forall o \in f : P, T \quad \text{on} < t, f, o > \text{is a roller} \tag{3.5}$$

Theorem 10. A set of tiles $t_i \in RA$ in a reachable area for a rolling pair P,T is stable on a tile $t \in T$ if

$$\forall f \in P \quad \text{s.t.} \quad |f| == |t|, \quad \forall o \in f : \forall t_i \in RA : \exists f_i, o_i \quad \text{s.t.} \quad < t, f, o > \sim < t_i, f_i, o_i > \tag{3.6}$$

Chapter 4

Implementation

The proof algorithm was implemented in python 3.8 and is available on github. It uses numpy and sympy for creating the minimal linearly independent base, and pygame to produce images.

4.1 Polyhedron format

Polyhedron's nets are represented by the dual graph of faces and their adjacent neighbours in clockwise order.
Platonics, some Archimedeans were generated using a python script named wrl_to_dict my promotor provided, used on .wrl files from [30]

```
johnson_nets['j1'] = {
0: [1, 2, 3, 4],
1: [0, 4, 2],
2: [0, 1, 3],
3: [0, 2, 4],
4: [0, 3, 1]}
#Example of implementation of a polyhedron's dual net graph
```

Johnson solids were generated using wrl_to_dict using .wrl files from [31]
Due to formatting issues, most Archimedeans, Prisms and Antiprisms were generated using a custom-made drawing tool (NetMaker.py) and based on net unfolding images from Wikipedia's

page[32]. Archimedeans were then checked by printing them out to verify the face connections.

Figure 4.1: Archimedeans which were printed out to verify edge correctness

4.2 Tiling format

Tilings are represented by the dual graph of the cells and their adjacent cells in clockwise order in a supertile. In practice, it is a python dictionary with face number as keys, and neighbour number and identifier tuple as value. The inside of a supertile is defined by all the face links with identifier 0 that are not linked to itself. All external links are paired by the face number and either (in order of priority) the same identifier but negative or the same identifier and can be zero if a face is linked to itself.

```
biisogonal_tilings['(3^3x4^2;4^4)1'] = \
{   0: [(2, 0), (0, 1), (1, 0), (0, -1)],
    1: [(0, 0), (1, 4), (3, 0), (1, -4)],
    2: [(0, 0), (3, 2), (3, 3)],
    3: [(1, 0), (2, 2), (2, 3)]}
#Example of implementation of a tiling's dual graph
```

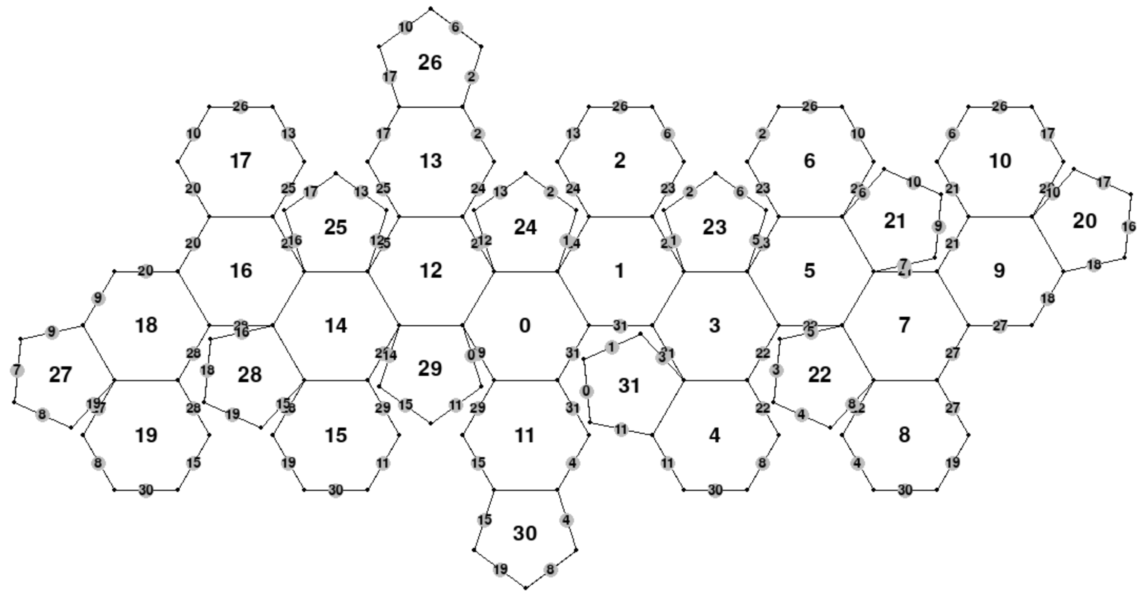


Figure 4.2: Example of screen of NetMaker.py

k-isogonal tilings were generated using a custom drawing tool (TilingMaker.py) based on tiling images from [9]. They were then checked for mistakes by visualising the tilings as images.
Tessellation tilings were generated manually based on tiling images from [21].
As explained earlier, the representation method is limited to periodic tilings, since it is defined as a supertile. Moreover, for every tiling generated by hand, a parallelogon-type supertile that can be used to tile the plane only by translations was purposely chosen.

4.3 Polyhedron face symmetries format

For optimisation reasons, Polyhedron faces that have exploration isomorphisms are described as a list of tuples containing (Face, Orientation) tuples that are equivalent symmetrical and are interchangeable during rolling. It is used by replacing matching entries by the smallest entry in the list, considered the "canon" representation of the symmetry class. Pairs with no equivalence form a lone tuple class. It is not used when face information should be maintained, but it greatly speeds up the search for reachable area's type.

```
'j1': [((0, 0), (0, 1), (0, 2), (0, 3)),
       ((1, 0), (2, 0), (3, 0), (4, 0)),
```

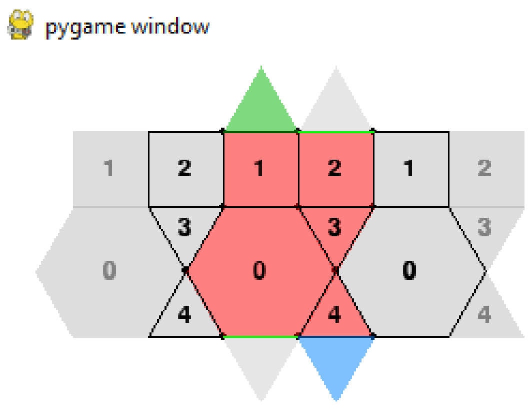


Figure 4.3: Example of screen of TilingMaker.py

```
((1, 1), (2, 1), (3, 1), (4, 1)),
((1, 2), (2, 2), (3, 2), (4, 2))]
```

Symmetries were generated using the method described in chapter 8 by looking at isomorphisms of the modified net dal graph of the polyhedron.

4.4 Translation symmetry vectors

The set of linearly independent vectors was built when exploring repeated states in the N-sized area by swapping newly encountered vectors in and out a list and checking linear dependence with sympy.Matrix.rref which returns the number of linearly independent columns.

4.5 Tiles and Polyhedron faces rolling graph implementation

When rolling on tilings that have no integer coordinates, a position was implemented as $<(point), c, f, o>$, point being the center of the polygon of the tile on the screen rounded. The rounding value is an integer to avoid floating comparison errors.
Exploration was done by pushing (cell, p1, p2, face, orientation) tuples on a queue and popping them. p1, p2 would represent the two first points of the n-gon defined by the face, which would then be re-oriented based on the orientation index. The cell's neighbour and face neighbour would need to be aligned by matching them backwards. Visited areas would be determined by a dictionary

containing the center of the n-gon of the visited tile as key, and a list of (cell, face, orientation) that reached it.

4.6 Supertile and super-position graph implementation

Super-positions were defined as a list of sets of states. Each set containing the compatible $< c, f, o >$ states.
Index of those sets were used in another list to describe changes between classes.
Tile coordinates were explored by matching a neighbouring state with a direction in the cartesian coordinates.
Exploration was done by pushing a (superstate identifier, x,y) tuple on a queue. Since moving on a parallelogon tiling only involves translations, there were no need for managing orientations.

4.7 Output

The script outputs images for face-complete tiles and stable tiles, and a pickled python dictionary containing rolling area with all the states to use for processing in other scripts.
It also puts a label on the rolling pair. A pairing was labelled "roller" if at least one of its connected state groups generated a rolling graph that covered the whole space. A pairing was a hollow plane roller if at least one of its connected state groups had two symmetry vectors. Out of those, a pairing was labelled "quasi-roller" if one group covered the whole space minus incompatible faces. If there was only one or zero linearly independent vectors, the pairing was labelled either band roller or bounded roller. Pairing that had no polygon in common were labelled incompatible. Rollers/quasi-rollers were labeled "stable" if every connected component was a roller/quasi-roller.
More details on the implementation can be found in the annex chapter 8 and Development history in section 9.

Chapter 5

Experimental results

We looked for rolling information in the following data:

All of the 122 relevant regular-faced polyhedrons.

- The 5 Platonic solids: tetrahedron, cube, octahedron, dodecahedron, icosahedron
- The 13 Archimedean solids: truncated tetrahedron, cuboctahedron, truncated cube, truncated octahedron, rhombicuboctahedron, truncated cuboctahedron, snub cube, icosidodecahedron, truncated dodecahedron, truncated icosahedron, rhombicosidodecahedron, truncated icosidodecahedron, snub dodecahedron
- The 92 Johnson solids: $j1 \longrightarrow j92$
- 12 Prisms and antiprisms: triangular prism, pentagonal prism, hexagonal prism, octagonal prism, decagonal prism, dodecagonal prism, square antiprism, pentagonal antiprism, hexagonal antiprism, octagonal antiprism, decagonal antiprism, dodecagonal antiprism

The data is available here.
Prisms and antiprisms using faces that cannot appear in tilings were ignored, as they could not be plane rollers (they could be band-rollers on tiling with a line of squares but this result is evident).
Notably missing chiral variations of J44, J45, J46, J47, J48.
The nets of the polyhedrons were sourced from various sources listed in 4

The 70 first tilings of the tiling catalog [9]:

- The 3 Platonic tilings: triangular grid, square grid, hexagonal grid
- The 8 archimedean tilings
- The 20 2-isogonal tilings
- The 39 3-isogonal vertex-homogenous tilings

The data is available here. There remains 106 tilings listed in the catalog that were not transcribed.

The RollingProof.py executing the rolling proof algorithm was ran for every pairing. The runtime for processing the 122 polyhedrons and 70 tilings is 9 minutes and half on an Intel Core i7-9750H CPU @ 2.60GHz with 15GB of usable RAM.

We found in our search set:

- Plane Rollers pairs : 33 stable, 62 unstable (total: 95 Rollers)
- Quasi-Plane Rollers : 43 stable, 120 unstable (total: 163 Quasi-Plane Rollers)
- Hollow Plane Roller : 199
- Band Rollers : 1115
- Bounded Rollers : 6579
- Incompatible pairs : 389

5.1 Result files online

A Google Drive folder is made accessible with various complementary data that couldn't fit in the thesis.

5.2 Recapitulative matrix of all pairings

A complete and exhaustive review of the findings is represented in matrix form in fig. 5.1.

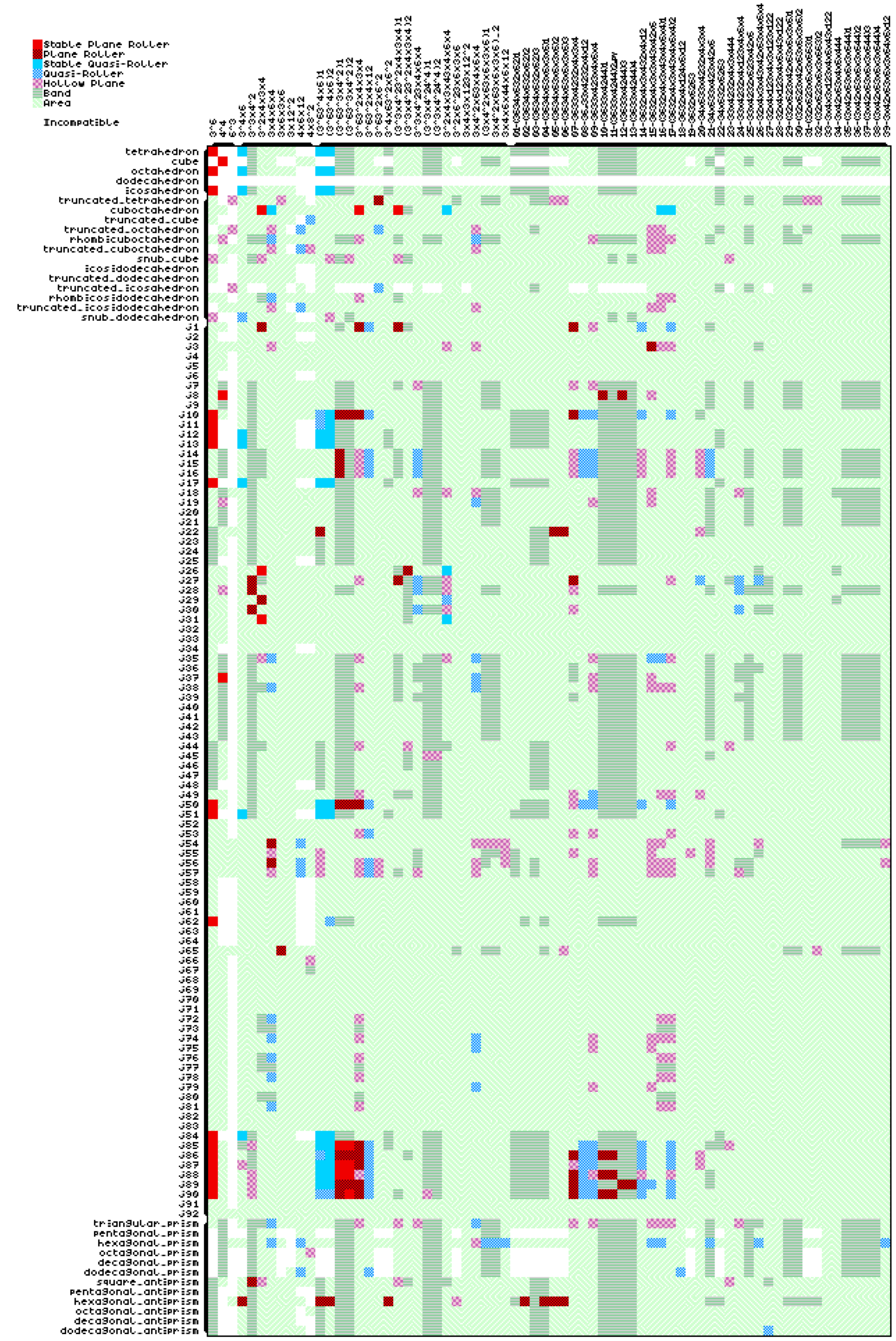


Figure 5.1: All rollers data

5.3 Condensed plane rollers results table

Condensed version of the previous result focusing on plane rollers.
Each entry links to a more detailed table in the annex.

Platonic Tilings (3)	Stable Plane Roller
3^6	tetrahedron, octahedron, icosahedron, j10, j11, j12, j13, j17, j50, j51, j62, j84, j85, j86, j87, j88, j89, j90
4^4	cube, j8, j37

No roller: 6^3

Archimedean Tilings (8)	Stable Plane Roller	Unstable Plane Roller
3^4x6		hexagonal antiprism
3^3x4^2		j27, j28, j30, square antiprism
$3^2x4x3x4$	cuboctahedron, j26, j31	j1, j29
$3x4x6x4$		j54, j56
$3x6x3x6$		j65

No roller: $3x12^2$, $4x6x12$, $4x8^2$

2-isogonals Tilings (20)	Stable Plane Roller	Unstable Plane Roller
$(3^6; 3^4x6)1$		j22, hexagonal antiprism
$(3^6; 3^4x6)2$		hexagonal antiprism
$(3^6; 3^3x4^2)1$	j85, j87, j88	j10, j14, j15, j16, j50, j86, j89, j90
$(3^6; 3^3x4^2)2$	j85, j87, j88, j90	j10, j50, j86, j89
$3^6; 3^2x4x3x4$	cuboctahedron	j1, j10, j50, j85, j86, j87, j89, j90
$3^6; 3^2x6^2$		truncated tetrahedron
$3^4x6; 3^2x6^2$		hexagonal antiprism

$(3^3x4^2;3^2x4x3x4)1$	cuboctahedron	j1, j27
$(3^3x4^2;3^2x4x3x4)2$		j26

No roller: $3^6;3^2x4x12$, $3^3x4^2;3x4x6x4$, $(3^3x4^2;4^4)1$, $(3^3x4^2;4^4)2$, $3^2x4x3x4;3x4x6x4$, $3^2x6^2;3x6x3x6$, $3x4x3x12;3x12^2$, $3x4^2x6;3x4x6x4$, $(3x4^2x6;3x6x3x6)1$, $3x4^2x6;3x6x3x6)_2$, $3x4x6x4;4x6x12$

3-isogonal vertex homogeneous (39)	Unstable Plane Roller
$02-(3^6;3^4x6;3^2x6^2)2$	hexagonal antiprism
$04-(3^6;3^4x6;3x6x3x6)1$	hexagonal antiprism
$05-(3^6;3^4x6;3x6x3x6)2$	j22, hexagonal antiprism
$06-(3^6;3^4x6;3x6x3x6)3$	j22, hexagonal antiprism
$07-3^6;3^3x4^2;3^2x4x3x4$	j1, j10, j27, j86, j88, j89, j90
$10-(3^6;3^3x4^2;4^4)1$	j8, j86, j88, j90
$11-(3^6;3^3x4^2;4^4)2.py$	j86, j88, j90
$12-(3^6;3^3x4^2;4^4)3$	j8, j89
$13-(3^6;3^3x4^2;4^4)4$	j89
$15-3^6;3^2x4x3x3x4;3x4^2x6$	j3

No roller: $01-(3^6;3^4x6;3^2x6^2)1$, $03-(3^6;3^4x6;3^2x6^2)3$, $08-3^6, 3^3x4^2;3^2x4x12$, $09-3^6;3^3x4^2;3x4x6x4$, $14-3^6;3^2x4x3x4;3^2x4x12$, $16-(3^6;3^2x4x3x4;3x4x6x4)1$, $17-(3^6;3^2x4x3x4;3x4x6x4)2$, $18-3^6;3^2x4x12;4x6x12$, $19-3^6;3^2x6^2;6^3$, $20-3^4x6;3^3x4^2;3^2x4x3x4$, $21-3^4x6;3^3x4^2;3x4^2x6$, $22-3^4x6;3^2x6^2;6^3$, $23-3^3x4^2;3^2x4x3x4;4^4$, $24-3^3x4^2;3^2x4x12;3x4x6x4$, $25-3^3x4^2;3^2x6^2;3x4^2x6$, $26-3^2x4x3x4;3x4^2x6;3x4x6x4$, $27-3^2x4x12;3x4x3x12;3x12^2$, $28-3^2x4x12;3x4x6x4;3x12^2$, $29-(3^2x6^2;3x4^2x6;3x6x3x6)1$, $30-(3^2x6^2;3x4^2x6;3x6x3x6)2$, $31-(3^2x6^2;3x6x3x6;6^3)1$, $32-(3^2x6^2;3x6x3x6;6^3)2$, $33-3x4x3x12;3x4x6x4;3x12^2$, $34-3x4^2x6;3x4x6x4;4^4$, $35-(3x4^2x6;3x6x3x6;4^4)1$, $36-(3x4^2x6;3x6x3x6;4^4)2$, $37-(3x4^2x6;3x6x3x6;4^4)3$, $38-(3x4^2x6;3x6x3x6;4^4)4$, $39-3x4^2x6;3x6x3x6;4x6x12$

5.4 Rolling Proofs and stability

The actual images are in the annex in section 11.1 for pacing concerns.
Individual images proving the rolling area were produced for rollers, quasi-rollers and hollow-plane rollers. The images are included in section 11.1 in the table for stable and unstable rollers. It also includes a graph of stable tiles for unstable rollers. The other tables (quasi-rollers, hollow-plane rollers) were not included for space concerns, but a complementary PDF will be included which contains extra tables.

The polyhedron renders displayed are courtesy of Wikipedia. They are rendered by a collaborative effort of several users and are licensed under the Creative Commons Attribution-Share Alike 3.0 or CC BY-SA license. Attribution of authorship via Wikipedia is indicated on the images. These images are thus under the CC BY-SA 4.0 license.

The rolling graphs were generated by the last step of the proof generating algorithm. Since several graphs were generated per pairing, the first one was chosen to be included in the table. For plane rollers that have several connected groups generating a rolling area, I have verified that every graph is the same, but with internal face symmetries (separate from translation symmetries which account for different face and orientations).

The stability graphs were generated for unstable rollers and unstable quasi-rollers where every group to account for in the stability also contains every tile in the plane. Stability graphs were not generated for hollow-plane rollers, because the unreached places might not be at the same places depending on which state we start on in the connected group, which makes comparisons more complicated.

The choices for the net and face-orientation example were generated by taking the compatible face whose polygon appears the least in the polyhedron, of the smallest index. This is so that someone wanting to use it would have an easier time picking the right face.

5.5 Various Rolling area images

A large amount of rolling area images were also produced for all pairings using another method. This set includes images for bounded areas rollers and band rollers. Several thousands of images were generated, then looked at. Some visually interesting results were compiled at imgur.com/gallery/yLgVtMH. The program can be found at ScreenspaceRoller.py.

5.6 Tessellation polyhedron rolling

The images are available in the table section 11.2.
The same program ran to explore tessellation polyhedron rolling graphs. This version of the program was based on a different proof idea, that did not use the parallelogon simplification of the rolling graph, as a lot of tessellation polyhedron generate conway-type supertiles. However while I found experimentally a lower bound for getting reliable results, I could not prove it mathematically, so there is no strictly defined proof attached to those results.
The generated image changes color based on the number of different $< c, f, o >$ states that reach it. It is an earlier version of the idea of face-complete marking.
The CFO adjacency matrix column is also an earlier attempt to define stability and face-complete tiles. It indicates how many connected groups of positions there are, which can be an early indication of stability if there are very few groups and that they would cover all possible faces combinations. However, stability graph and outright marking face-complete tiles would end up giving clearer results. While visually pleasing, this earlier work is included to mark the progression of the visualisation of the information.
Turning the tilings that are defined trough conway-type supertiles back into parallellogon supertiles to run those pairings with the other program would be an easy task. But those tilings can also be found lower back in the tilings catalog, so the priority would rather be to extend the amount of tilings included from the catalog into the main program.

Condensed version of the table:

Tessellation polyhedron that are plane rollers on their tessellation tiling: Tetrahedron, Cube, Octahedron, Icosahedron, J8, J10, J12, J13, J17, J50, J51, J84, J86, J87, J88, J89, J90
Tessellation polyhedron that are band rollers on their tessellation tiling: Hexagonal Antiprism, J1, J14, J15, J16, J49

Chapter 6

Results discussion

We have listed 71 plane roller pairs, 105 quasi-rollers, and others.
If willing, all pairings can be used in various rolling games, but some pairings have properties that are more desirable for ease of use. In particular, having a lot of stable tiles, and a lot of tiles that can be reached with any face. In addition, the results show some pairings that have tiles reachable with any face and orientation, which could be a new wrinkle to add to puzzle games which is often ignored due to its complexity.

6.1 General remarks

We can see in section 5.2 that the more complex the tiling, the less likely there will be a polyhedron rolling it. However it is unlikely that we would find no more rollers at all by going down the list of tilings. The list has room to grow, as there are some tessellation polyhedron tilings that were not present in the initial list that allow a plane rolling.
Furthermore, even complex tilings have plenty of hollow-plane rollers and quasi-plane rollers, which are still interesting to study. In fact, in our subset, there are more quasi-rollers than any other type of plane roller. We might not discuss in too much details all quasi-rollers due to space concerns.
As shown in the gallery linked in section 5.5, some band rollers and bounded rollers have pretty intricate designs, and could still be used in puzzle games as well, by limiting the size of the playground.

6.2 Discussing stable plane roller cases

We will discuss interesting results about face-complete areas of stable rollers in the table of section 11.1. Please read the legend in section 11.1 to understand the images.
Stable rollers are rollers on every of their starting position.

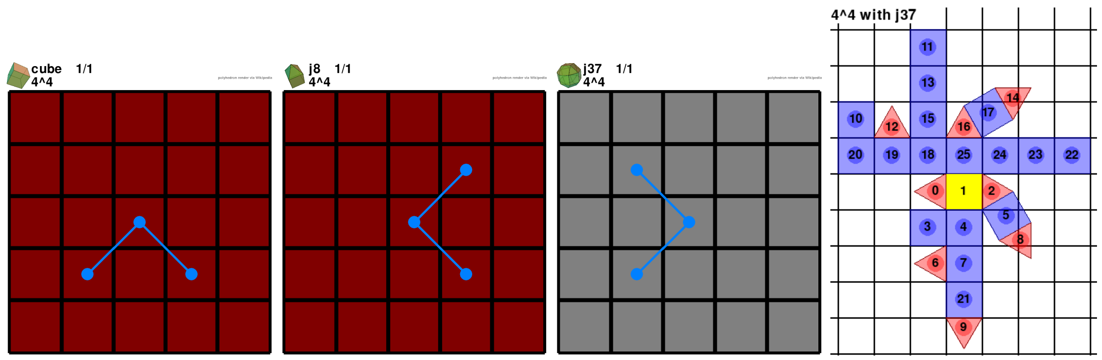


Figure 6.1: Cube, J8, J37 (and its net) rolling in the square grid

On the 4^4 square tiling: As expected, the cube and J8 (which is basically a cube with a missing face) can roll the square tiling with every face. But not with every orientation. The cube for example can only do 180° rotations on its faces. The only other candidate, J37, cannot freely reach every tile with all of its faces.

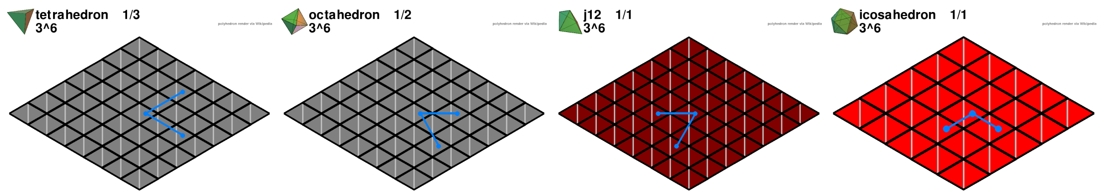


Figure 6.2: Tetrahedron, Octahedron, J12 and Icosahedron rolling in the triangular grid

On the 3^6 triangular tiling: Tetrahedron, Octahedron would probably be anyone's first guess of a roller. However, for rolling puzzles, their rolling face positions are fixed by their starting position, no tile can be reached with every face, which limits the type of puzzles that are possible.
J12, J13, J90 are totally face-complete, for face-matching puzzles. Icosahedron, J10, J11, J17, J50, J51, J62, J84, J85,J86; J87, J88, J89 are face and orientation-complete, which could lead to even more advanced puzzles than what's possible with cubes on square tilings.

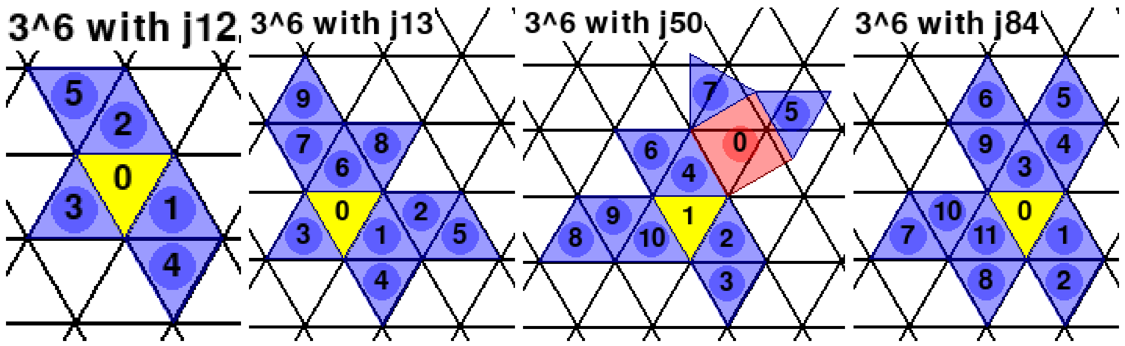


Figure 6.3: Net of J13, J13, J50 and J84

Out of those, J12 (triangular bipyramid) is the closest to the cube: similarly face-complete rolling, and similarly only 6 faces. J13 is not far behind with 10 faces for face-complete rolling. For face-orientation-complete rolling, the icosahedron is an interesting pick, as it is the only regular polyhedron here, despite its 20 faces. The smallest picks would be J50 with 10 compatible faces (and one incompatible). The incompatible face could be interesting to play around as it blocks some paths. Without incompatible faces, J84 (snub disphenoid) with 12 faces is the second smallest.

On non-regular tilings: Out of the following tilings that were inspected, only $(3^6; 3^3 * 4^2)_1$ and $(3^6; 3^3 * 4^2)_2$ are consistently face and orientation-freerolling and stable. In fact, all of their stable rollers in the table are face-orientation-complete, except for J90.
For $(3^6; 3^3x4^2)1$: J85 J87 J88
For $(3^6; 3^3x4^2)2$: J85 J87 J88 J90

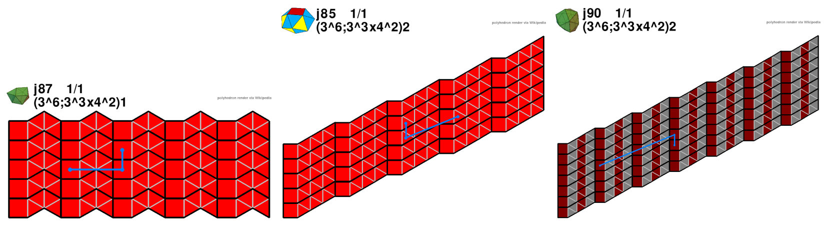


Figure 6.4: $(3^6; 3^3x4^2)1$ (rolled with J87), $(3^6; 3^3x4^2)2$ (rolled with J85) and $(3^6; 3^3x4^2)2$ again (rolled with J90)

The remaining three tilings that have a stable roller, $3^2x4x3x4$, $3^6; 3^2x4x3x4$, $(3^3x4^2; 3^2x4x3x4)1$, interestingly are all rolled by the cuboctahedron. Maybe its structure is particularly suited for rolling. Like shown, $3^2x4x3x4$ has no face-complete tile, while the other are face-complete for all of their squares only.

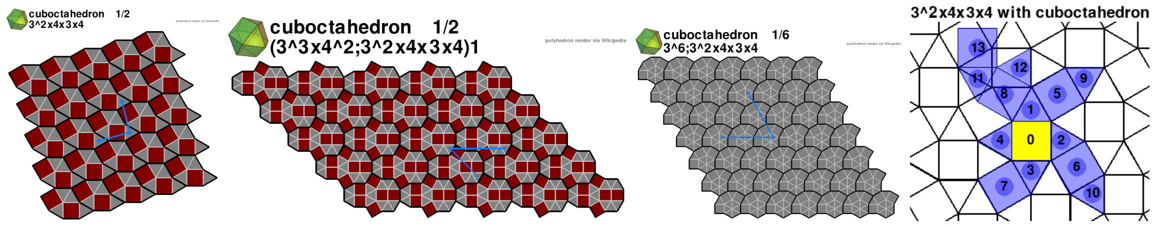


Figure 6.5: The three remaining tilings with stable roller are rolled by the cuboctahedron (net)

But even rolls that are a mix of reachable tiles and face-complete tiles can be used for puzzles. This means that face-matching conditions could be preferably set on the face-complete tiles. Since this section only deals with stable rolling, the starting position only changes the orientations of the faces, not their reachability. I also checked that the stability of face-complete positions doesn't change by comparing the images of all the generated groups.
Tiles that are not face-complete can still be used for face-matching, if the puzzlemaker is careful to pick one of the faces that is guaranteed to be reached based on the starting position.

6.3 Discussing unstable plane roller cases

Unstable cases are not rollers on some of their starting positions. Their stable tiles are highlighted in dark grey in the "stability" column. They are still rollers so every tile has a starting position that belongs to a plane rolling, but unless you have a list of working $< t, f, o >$, you are better off starting from a stable position.

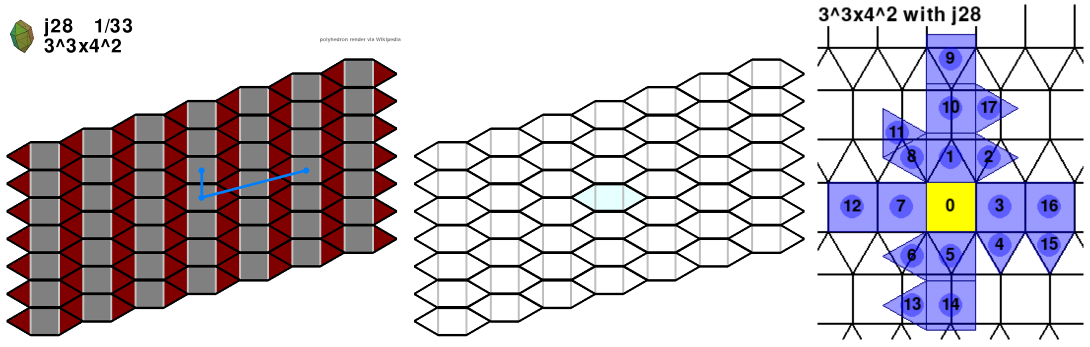


Figure 6.6: Roller with no stable position. The net view helps positioning it.

Some pairing like j28 with $3^2x4x3x4$ have no stable tile, which means that using it to roll the space, we have to be careful to choose a working starting position, for example the face and tile and orientation used in the net image.
Let's highlight some unstable pairing of tilings that can be rolled by the simplest of the Johnson solids, the square pyramid J1. $3^2x4x3x4$ makes a comeback. J1 also rolls in $3^6;3^2x4x3x4$, $(3^3x4^2;3^2x4x3x4)1$ and $3^6;3^3x4^2;3^2x4x3x4$. And they are all conveniently stable on square tiles. Since J1 is a small and manageable polyhedron, and with the convenience of the square tiles being stable, those tilings might get some use in puzzles.

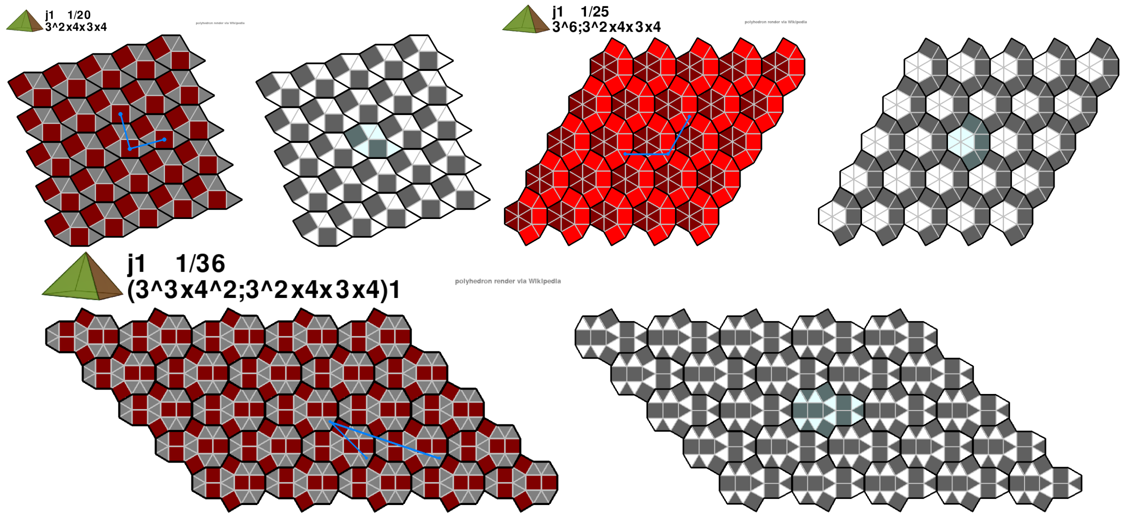

j1 1/20
3^2x4x3x4
polyhedron render via Wikipedia
j1 1/25
3^6;3^2x4x3x4
polyhedron render via Wikipedia
j1 1/36
(3^3x4^2;3^2x4x3x4)1
polyhedron render via Wikipedia

Figure 6.7: The tessellations that roll the square pyramid, and their stable tiles

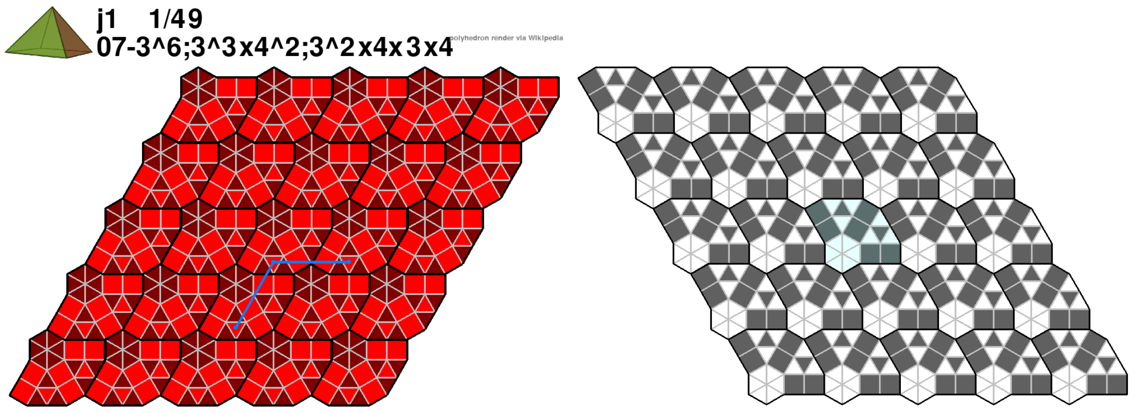


Figure 6.8: J1 on $3^6;3^3x4^2;3^2x4x3x4$

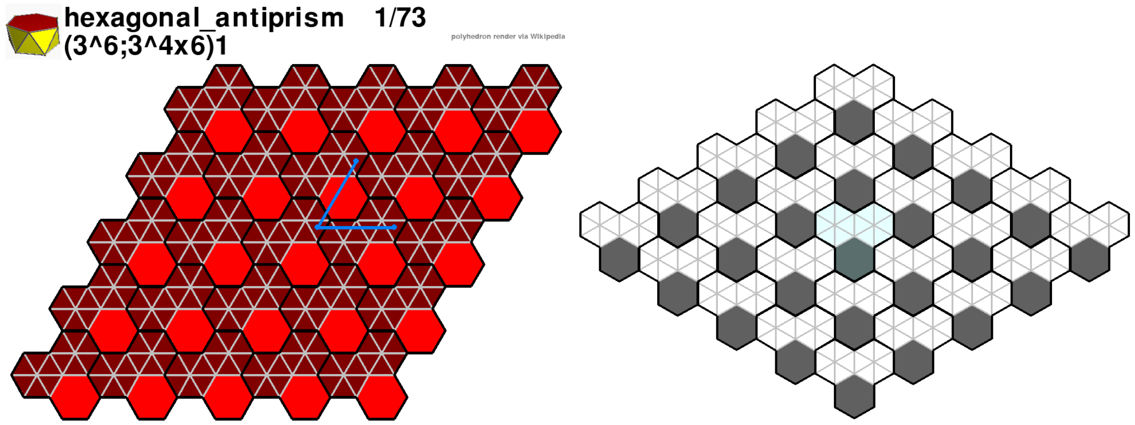


Figure 6.9: Hexagonal antiprism on $(3^6;3^4x6)1$

For rolling pairs that are at least face-complete everywhere, the hexagonal antiprism on $(3^6;3^4x6)1$ stands out. Especially since the hexagonal tiles are also orientation-complete, maybe making for interesting multilayered puzzles where several antiprisms have to be fitted in their corresponding orientation.
There are many unstable rolls for $(3^6;3^3x4^2)1$ and $(3^6;3^3x4^2)2$ that are stable on square only but are at least face-complete everywhere. In fact, their orientation-complete tiles are always stable.

Theorem 11. Since starting on a stable tile with any compatible face and orientation guarantees to roll the space, and face-orientation-complete tiles can be reached with any compatible face and orientation in a specific rolling, face-orientation-complete tiles are by definition stable.

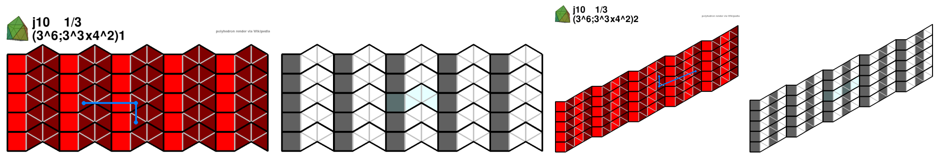


Figure 6.10: $(3^6;3^3x4^2)1$ and $(3^6;3^3x4^2)2$ once again share polyhedrons and rolling patterns

As for other notable tilings, $3^6;3^2x4x3x4$ does roll with other shapes than J1, but only has both face-complete tiles and stable tiles with J10 and J90.

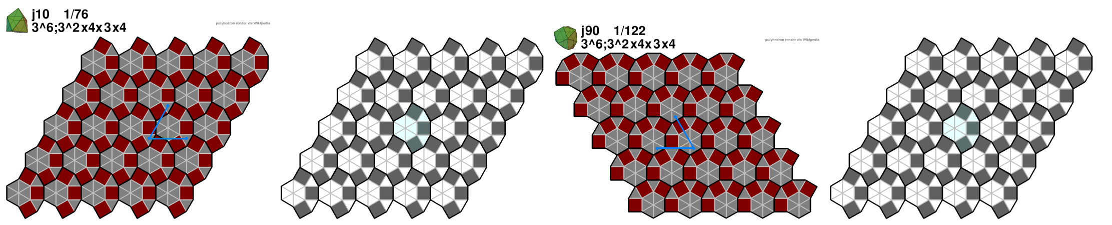


Figure 6.11: $3^6; 3^2x4x3x4$ with J10 and J90

Lastly, J27 on $(3^3x4^2; 3^2x4x3x4)1$ has a very particular, if not unwieldy, rolling pattern, with strange stability tiles, showing that stable tiles are not always face-complete. It is otherwise unremarkable.

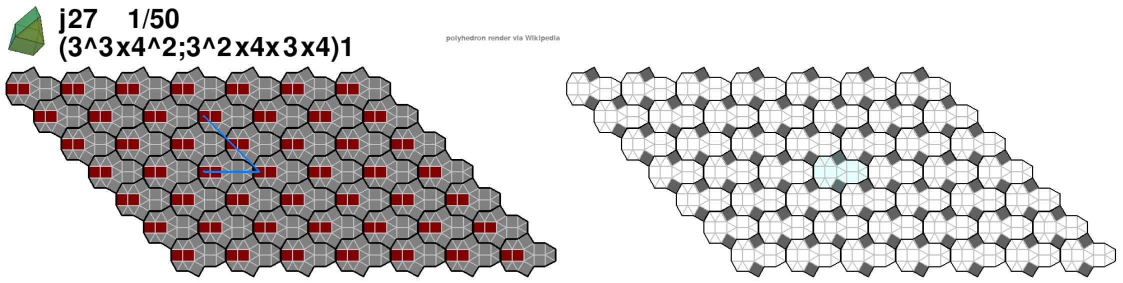


Figure 6.12: J27 on $(3^3x4^2; 3^2x4x3x4)1$

6.4 Discussing some hollow plane rollers that are quasi-rollers

Note. Quasi-plane rollers and hollow-plane rollers tables are not included in this work. They are available to browse in the Google Drive folder.

We have reviewed the most interesting plane rollers, but there are still more to explore. Quasi-rollers are tiling, polyhedron pairs for which we deliberately ignore incompatible tiles.

When using a pairing that rolls on the plane, one of the issues is to chose a starting and ending position that can be reached and reach the plane. If some tiles cannot be reached, they should be avoided or they would cause problem during that choice. However, for quasi-rollers it is impossible to accidentally choose a wrong position because the unreached tiles are incompatible with the polyhedron's faces. Hence the name quasi-roller.
Since there are many, many quasi-rollers (so many that we didn't include a table in this thesis),

and a lot of them look similar to each-other (due to most just missing an hexagon), we will focus on making broad general statements.

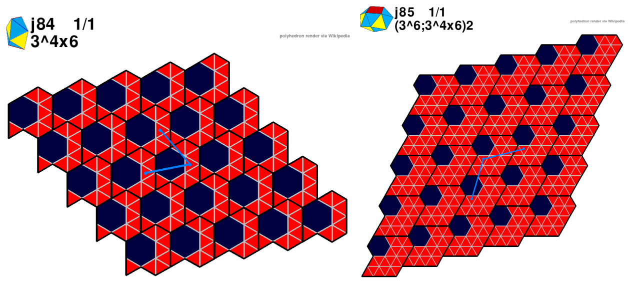


Figure 6.13: j84 on 3^4x6, j85 on $(3^6;3^4x6)2$. Most stable quasi-rollers look like that.

There are 18 stable quasi-rollers that are face-orientation complete except for the missing hexagon or octahedron featured in the tiling. 5 are face-complete.
Most of the unstable quasi-rollers are the same, with notable difference that some are missing triangles, squares, octagons or dodecagons. It makes up the lack of puzzle useability in variety of tiling patterns and gives prisms with large faces a time to shine.

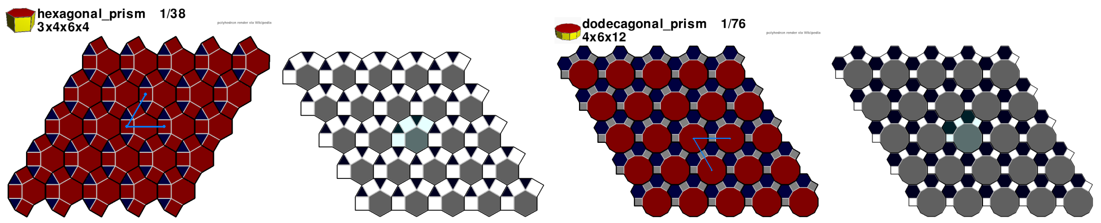


Figure 6.14: hexagonal prism on $3x4x6x4$, dodecagonal prism on $4x6x12$

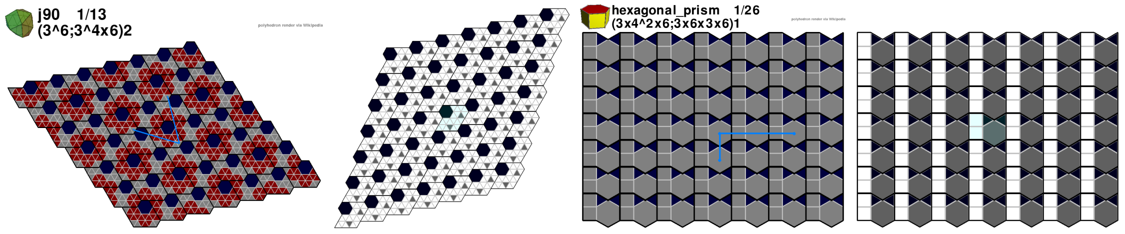
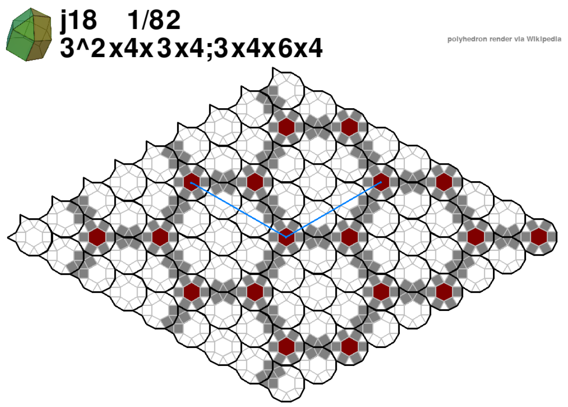


Figure 6.15: j90 on $(3^6;3^3x4^2)2$, hexagonal prism on $(3x4^2x6;3x6x3x6)1$

6.5 Discussing some hollow plane rollers

Non-rollers tables are not included in this work. However, you can look at a gallery for hollow-plane rollers here: imgur.com/a/e4Cs493 and a gallery for select band rollers and bounded rollers here imgur.com/gallery/yLgVtMH

Non-rollers are not simple to use in the context of puzzles. No stability, weak face matching properties. But their rolling patterns are often intricate and very pleasing to look at. Rather than puzzles, those might be more adapted for action games with puzzle elements, dynamically showing the available path, for controlling non-player elements.



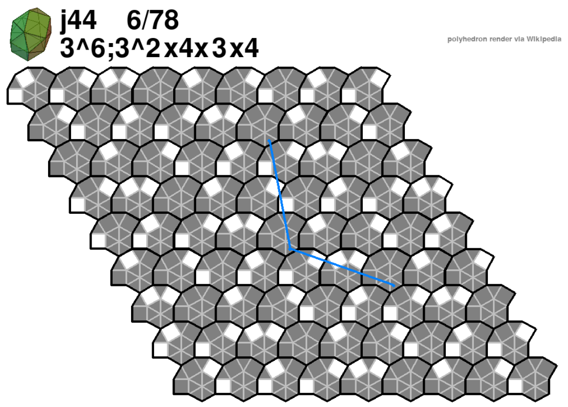
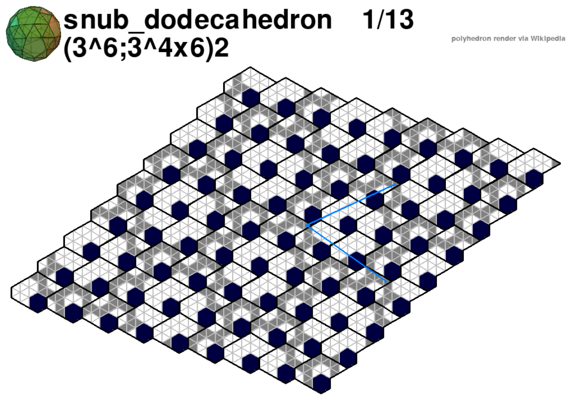


Figure 6.16: j18 on $3^2x4x3x4; 3x4x6x4$, j44 on $3^6; 3^2x4x3x4$



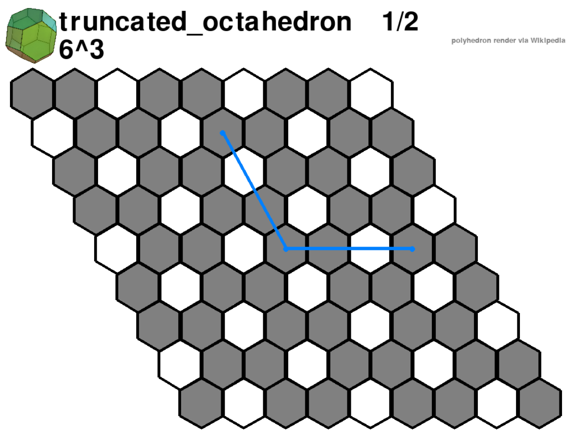
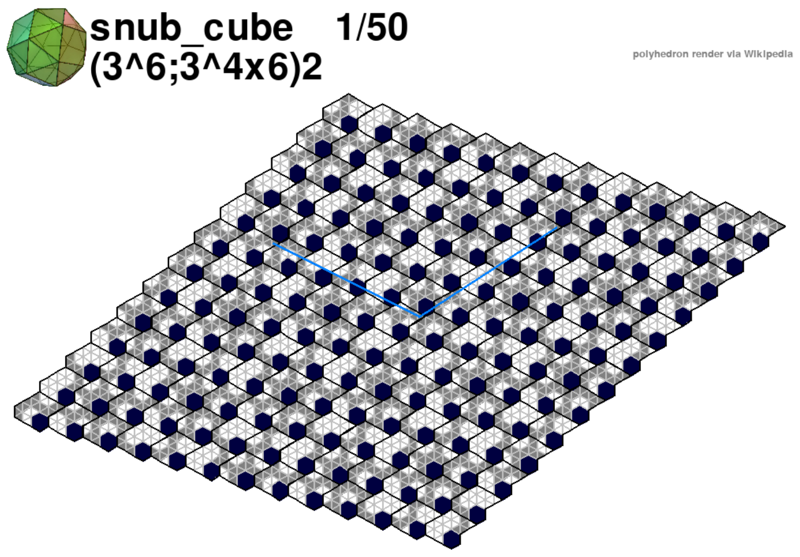


Figure 6.17: snub dodecahedron on $(3^6; 3^4x6)2$, truncated octahedron on 6^3



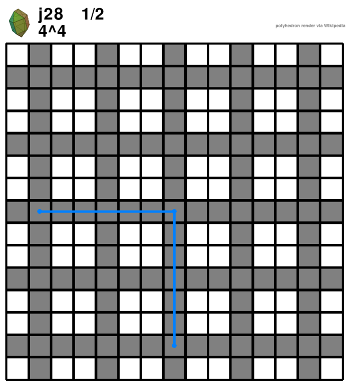


Figure 6.18: snub cube on $(3^6; 3^4x6)2$, j28 on 4^4

6.6 Limitations

Limited number of tilings tested. As mentionned in section chapter 5, while we have the exhaustive list of regular-faced polyhedrons, we do not have a complete list of tilings. Currently tilings are added manually, which explains the slow rate and small pool.
[9] does explain a way to generate tilings, at it might get implemented in the future to allow for more extensive searches.
Another possibility is to generate the tilings from the polyhedron themselves.

Only periodic tilings. Another issue is that aperiodic and non-periodic tilings are not part of the search set. Some of those tilings have a clear structure, such as radially-symmetrical tilings compiled in [11], or hyperbolic tilings as shown in HyperRogue or [12]. Exploring such spaces is still a graph connectivity problem. Generating a rolling proof in those spaces might amount to the same logic of describing some form of symmetry, isolate looping parts of the space and explore those independently. A good coordinates system would be helpful.

Better representation of symmetries in shapes/tilings. As of now, the symmetry vectors are for face-orientation repetitions, not the visible structure of the rolling graph. A separate set of symmetry vectors that accounts for the polyhedron's internal symmetry and the tiling's symmetry could help clean up the visualisations even more. We also don't have a visual representation for partially symmetrical orientations, like the cube that allows 180° rotations for its faces on 4^4.

Stable on the path. Stability graphs were not produced for hollow-plane rollers, band or bounded rollers. However, limited stability could be defined as the states that produce the same rolling area, even when not connected.

Non-regular-faced rolling The described algorithm could be rewritten to function on non-regular-faced polyhedron and tilings. But some complications in the proof might occur.

Chiral version Several regular-faced polyhedron have chiral variations that are missing. Namely, J44, J45, J46, J47, J48. Since the tilings can be interpreted in two ways chirally as well, the simplest is to add those polyhedron variations in the list.

Chapter 7

Conclusion

What is a roller?

First, an extension of the concept of rolling cubes for every polyhedron and tiling combinations. We defined many terms in chapter 2 but ultimately, the physical or virtual manipulation of a shape on a grid is the most telling.

Second, a graph connectivity problem exploring two different systems at the same time - the polyhedron faces and the tiles on the tessellation, or the super-positions space and the supertile coordinates - a bounded space and an infinite one. In chapter 3 we wrote an algorithm to prove that a rolling pair is a plane roller, by transforming regular-faced tilings into a parallelogon supertile structure allowing for using symmetry vectors of the rolling areas on a cartesian plane.

Third, beautiful images of reachable areas in the space. In chapter 5 we outlined a method to differentiate plane rollers, quasi-plane rollers, band rollers, bounded rollers and incompatible rollers. We saw many patterns, simple and complex, and tried to make sense of them.

Fourth, puzzle pieces. In chapter 6 we looked at specific rolling pairs that could be interesting for puzzle games. We have discussed the limitations of the cube roller. We highlighted, J12, the triangular bipyramid, on the triangular tiling, which seemed to give the most similar results as the cube on the square tiling, as a stable, face-complete roller with few faces. Another shape, J1, could roll on four tilings as a plane roller with few faces and some stable tiles.

And more in the future. Limitations of the work were highlighted, mostly possibilities in expanding the amount of tilings considered. My hope is that this work will continue on in different puzzlemaker circles to encourage the use of alternative polyhedron and tilings.

Bibliography

[1] Johannes Kepler. *Harmonices Mundi.* II. 1619.

[2] Matila Costiescu Ghyka. *The geometry of art and life.* 1946.

[3] Darrah Perry Chavey. “PERIODIC TILINGS AND TILINGS BY REGULAR POLYGONS (TESSELATION)”. PhD thesis. The University of Wisconsin-Madison, 1984.

[4] Helmer Aslaksen et al. “In Search of Demiregular Tilings”. In: *Sarhangi, R. y Sharp, J. Bridges London: Mathematical Connections in Art, Music, and Science. Londres, Tarquin Books* (2006), pp. 533–536.

[5] Branko Grünbaum and Geoffrey C Shephard. “Tilings with congruent tiles”. In: *Bulletin of the American Mathematical Society* 3.3 (1980), pp. 951–973.

[6] Branko Grünbaum and GC Shephard. “The ninety-one types of isogonal tilings in the plane”. In: *Transactions of the American Mathematical Society* 242 (1978), pp. 335–353.

[7] Branko Grünbaum and Geoffrey C Shephard. “The eighty-one types of isohedral tilings in the plane”. In: *Mathematical Proceedings of the Cambridge Philosophical Society.* Vol. 82. 2. Cambridge University Press. 1977, pp. 177–196.

[8] Branko Grünbaum and Geoffrey C Shephard. “Tilings by regular polygons”. In: *Mathematics Magazine* 50.5 (1977), pp. 227–247.

[9] Darrah Chavey. “Tilings by Regular Polygons—II A Catalog of Tilings”. In: *Symmetry 2* (1989), pp. 147–165.

[10] H. C. Williams. “Tilings and patterns, by B. Grunbaum and G. C. Shephard. Pp 700. £54·95. ISBN 0-7167-1193-1 (hardback) (Freeman)”. In: *The Mathematical Gazette* 71.458 (1987), pp. 347–348. DOI: 10.2307/3617109.

[11] Paul Hofmann. 2019. URL: https://www.biglist-tilings.com/.

[12] Eryk Kopczyński, Dorota Celińska, et al. *Hyperbolic grids and discrete random graphs.* Tech. rep. 2017.

[13] Euclid. *Elements.* I.

[14] Judith V Field. “Rediscovering the Archimedean Polyhedra: Piero della Francesca, Luca Pacioli, Leonardo da Vinci, Albrecht Dürer, Daniele Barbaro, and Johannes Kepler”. In: *Archive for History of Exact Sciences* 50.3/4 (1997), pp. 241–289.

[15] B Grünbaum and NW Johnson. “The faces of a regular-faced polyhedron”. In: *Journal of the London Mathematical Society* 1.1 (1965), pp. 577–586.

[16] Norman W Johnson. “Convex polyhedra with regular faces”. In: *Canadian Journal of mathematics* 18 (1966), pp. 169–200.

[17] Viktor Abramovich Zalgaller. “Convex polyhedra with regular faces”. In: *Zapiski Nauchnykh Seminarov POMI* 2 (1967), pp. 5–221.

[18] Yacine Chitour et al. “Rolling polyhedra on a plane, analysis of the reachable set”. In: *Algorithmic Foundations of Robotics II* (1997), pp. 277–285.

[19] Antonio Bicchi, Yacine Chitour, and Alessia Marigo. “Reachability and steering of rolling polyhedra: a case study in discrete nonholonomy”. In: *IEEE Transactions on Automatic Control* 49.5 (2004), pp. 710–726.

[20] Jin Akiyama. “Tile-makers and semi-tile-makers”. In: *The American Mathematical Monthly* 114.7 (2007), pp. 602–609.

[21] Jin Akiyama et al. “Determination of all tessellation polyhedra with regular polygonal faces”. In: *International Conference on Computational Geometry, Graphs and Applications*. Springer. 2010, pp. 1–11.

[22] K. Buchin and M. Buchin. “Rolling block mazes are PSPACE-complete”. English. In: *Journal of Information Processing* 20.3 (2012), pp. 719–722. ISSN: 0387-6101. DOI: 10.2197/ipsjjip.20.719.

[23] Markus Holzer and Sebastian Jakobi. “On the complexity of rolling block and Alice mazes”. In: *International Conference on Fun with Algorithms*. Springer. 2012, pp. 210–222.

[24] Jiawei YAO. “Research on the Minimum Moves of Rolling Cube Puzzles”. In: (2021).

[25] Kevin Buchin et al. “On Rolling Cube Puzzles.” In: *CCCG*. 2007, pp. 141–144.

[26] Craig S Kaplan. “Introductory tiling theory for computer graphics”. In: *Synthesis Lectures on Computer Graphics and Animation* 4.1 (2009), pp. 1–113.

[27] Doris Schattschneider. “Will it tile? Try the Conway criterion!” In: *Mathematics Magazine* 53.4 (1980), pp. 224–233.

[28] Ian Gambini and Laurent Vuillon. “An algorithm for deciding if a polyomino tiles the plane”. In: *RAIRO-Theoretical Informatics and Applications* 41.2 (2007), pp. 147–155.

[29] Amit Patel. *Red Blob Games: Hexagonal Grids*. Mar. 2013. URL: https://www.redblobgames.com/grids/hexagons/.

[30] *The Platonic and Archimedean Solids (VRML)*. Apr. 1996. URL: http://www.geom.uiuc.edu/software/weboogl/zoo/polyhedra.wrl.html.

[31] Vladimir Bulatov and George W. Hart. *Johnson solids (VRML format)*. 1996. URL: http://bulatov.org/polyhedra/johnson/index_vrml.html.

[32] *Archimedean solid*. May 2021. URL: https://en.wikipedia.org/wiki/Archimedean_solid.

Part II

Annex

Chapter 8

Implementation discussion

8.1 Proving Polyhedron face-orientation equivalence classes

To speed up the search, we take advantage of internal isomorphisms in the polyhedron, as they would reach the same areas. We take the dual graph of the net (ordered graph of faces neighbours) and modify it to isolate one Face and one Orientation (starting neighbour). Then we compare it to another modified graph with another Face' and Orientation'. We compare the two graphs (using SageMath) and if they are Isomorphic, the two (Face, Orientation) pairs can be treated as the same class in our search.

Modification antennae

In practice we modify the graph by adding antennae of different lengths to the Face and neighbours. We need at least 3 antennae, one on the face considered and two on the neighbours to determine Orientation and Chirality. But we can have an antenna on every neighbour for the same result (experimentally tested).

For example we see that all 4 orientations of the square face on the square pyramid (0) generate isomorphic graphs. We can see that the square face and the triangular faces will create non-isomorphic graphs to eachother. The triangular faces will create separate isomorphic classes for each orientation (square face in front, square face on the left, square face on the right). So in total there are 4 [Face, OrientationIndex] equivalence classes for the Square Pyramid, as seen in the dictionary entry below:

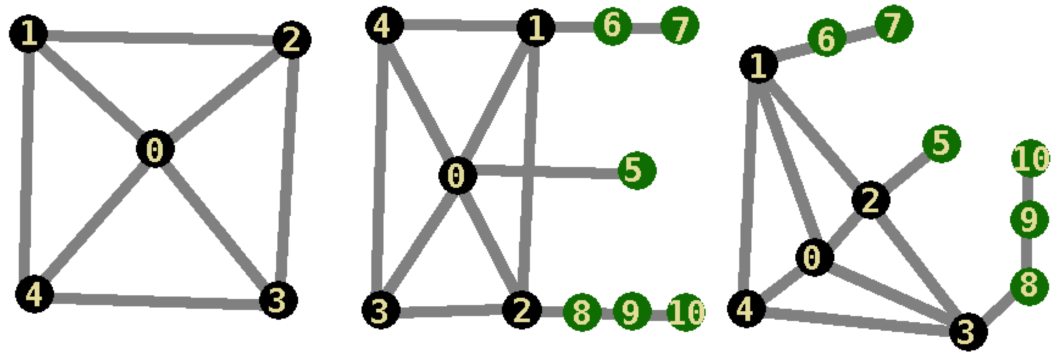


Figure 8.1: The square pyramid and its modifications, looking at Face 0 and Face 2 in the given Orientations and Chirality (which neighbours have the 2 and 3 antenna).

```
'j1': [((0, 0), (0, 1), (0, 2), (0, 3)),
       ((1, 0), (2, 0), (3, 0), (4, 0)),
       ((1, 1), (2, 1), (3, 1), (4, 1)),
       ((1, 2), (2, 2), (3, 2), (4, 2))]
```

Note We cannot use this optimisation if we care about maintaining unique Face/Orientation information (for example to distinguish perfect/partial roller).

Questions to consider:

Question Does it matter that is_isomorphic doesn't care about/doesn't maintain neighbour order? Is there a case, a polyhedron for which this is an issue? Does the order matter when representing a polyhedron as a dual graph, or will the result always be the same polyhedron?

Answer Polyhedral graph have unique dual graph (mathworld.wolfram.com/DualGraph.html) so changing the order of the neighbours wouldn't change the polyhedron, except for chirality (turning the graph inside-out or outside-in). This is fixed by having three distinct antennae that forces the graph to be in one chirality for comparisons.

Question What if we only use one sub-antenna?

Answer Orientations of opposite mirror-chirality will get confused.

Example For the square pyramid, if we only use one extra antenna, we have 3 classes instead of 4, because cannot differentiate left or right orientation of triangles, because order information is lost.

```
'j1':[((0, 0), (0, 1), (0, 2), (0, 3)),
      ((1, 0), (2, 0), (3, 0), (4, 0)),
      ((1, 1), (1, 2), (2, 1), (2, 2), (3, 1), (3, 2), (4, 1), (4, 2))]
```

Question What if we use two sub-antennae of the same length?

Answer SImilarly to the one-antenna question, there are cases where we cannot differentiate which Orientation we are talking about. The antennae are created in list order, but because is_isomorphic doesn't care about faces order, if the two antennae are not of different lenght, we lose chirality information, so for a mirror-symmetrical polyhedron, it might confuse two neighbouring orientations whose chosen neighbours are mirror-symmetrical to be equivalent, despite not even pointing towards the same direction.

Example For the square pyramid, if we use two extra antennae, but we don't differenciate them by length, we have 3 classes, because we cannot differentiate the orientation centered on the square face and the one centered on the previous neighbour, because order information is lost.

```
'j1':[((0, 0), (0, 1), (0, 2), (0, 3)),
      ((1, 0), (1, 2), (2, 0), (2, 2), (3, 0), (3, 2), (4, 0), (4, 2)),
      ((1, 1), (2, 1), (3, 1), (4, 1))]
```

Question Is one isomorphism test enough to deduce three different isomorphisms? (Just by swapping the three antennae's roles.)

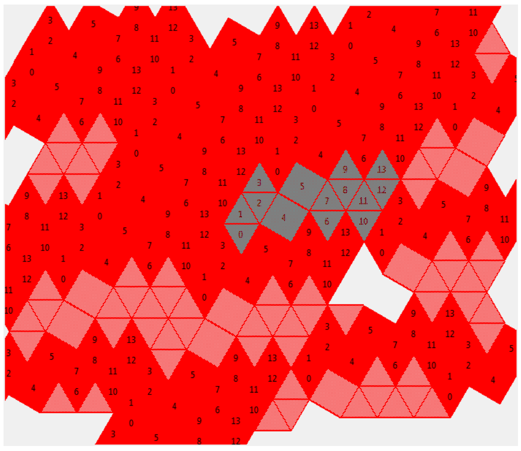
Answer Only if the neighbours are each other's neighbours themselves.

Chapter 9

Development history

9.1 shapesdraw.py (2017)

Written in Python in Tkinter in 2017 for the Computational Geometry course project
The project presentation can be seen here.



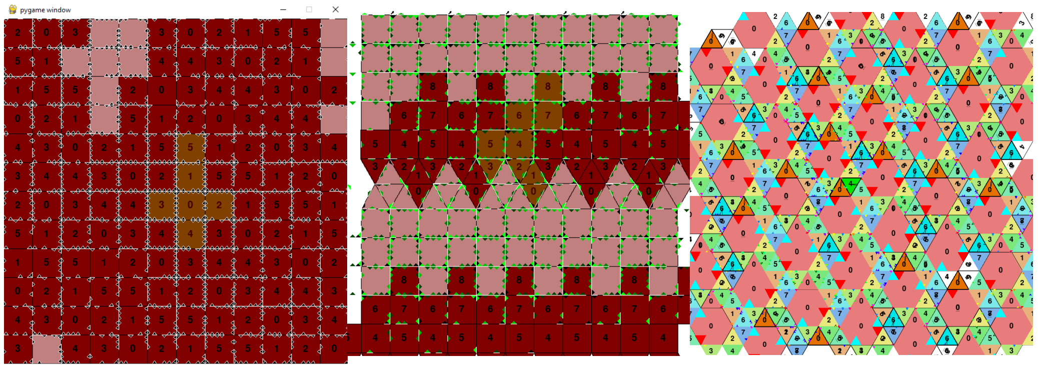
Red for parts reached with same $< c, f, p >$ as in the original supertile. Only contains 10 tessellation polyhedron painstakingly converted by hand. Recursive search call with a global keeping track of a list of **order-sensitive face neighbours per cell id**, which is equivalent to remembering the (face, orientation) pair later on. Since we are rolling in the net-tessellation, the search space is len(net)^{2}*sides.

This lead to the search finishing earlier than it can fill the screen, and holes in the visualisation.

9.2 main.py, NetDrawer.py (renamed NetMaker.py) (nov. 2020)

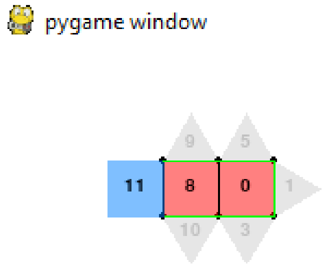
9.2.1 Upgrade to Pygame

Upgraded the visualisation from Tkinter to Pygame. Later on I added various ticks to keep track of rotations and face, which rendered the visualisation slower and harder to read. Each triangle on the side of a shape is a different face that can reach it, and this data is shown for each side=orientation. Color was added to try to cram more information into the visualisation.



9.2.2 NetDrawer.py

Created NetDrawer.py, a small tool to click to manually unfold the net of a given polyhedron. It would later evolve into more useful tools. It was to accellerate the tessellation net conversion to dictionary because a lot of the complex ones were tricky to convert by eye. Most of the tessellation polyhedron converted.



9.3 IsohedralTileRollExplore.py, CFO class matrix (dec. 2020)

9.3.1 IsohedralTileRollExplore.py, visualise mega-tile coordinates

Reimplementation of the exploration algorithm, but based on neighbouring mega-tiles, create a conway neighbours coordinates system, and color the explored tiles based on their mega-coordinates.

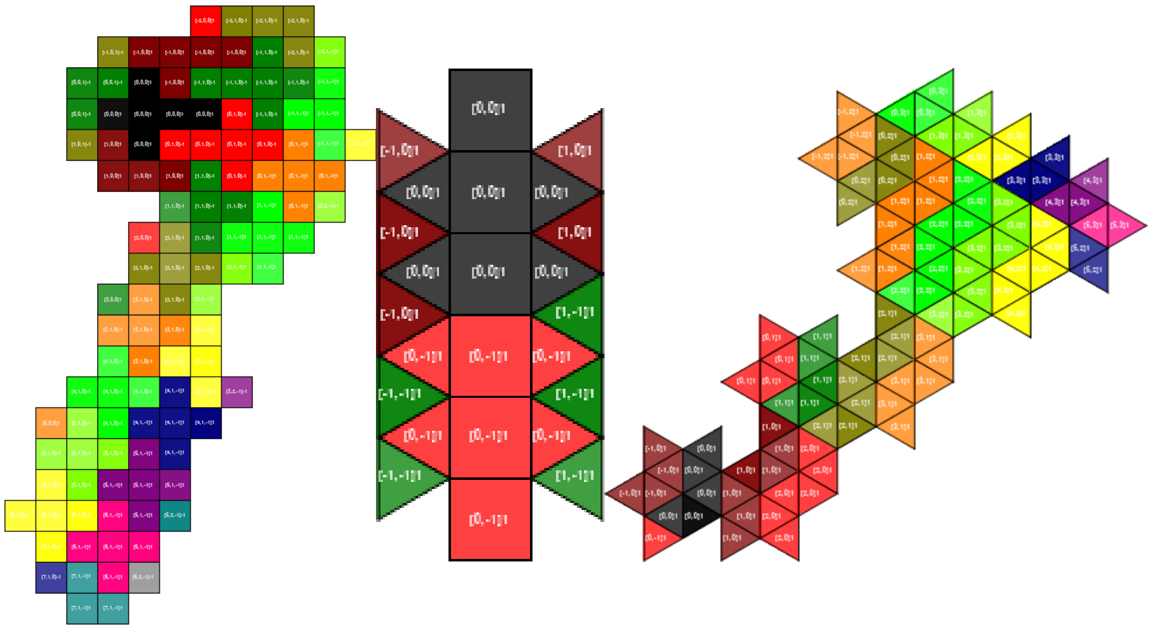
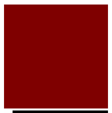
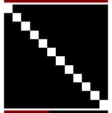


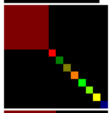
Figure 9.2: Exploration of space but colored by mega-tile coordinates

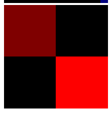
9.3.2 Case-Face-Orientation Classes Adjacency Matrices

Based on that coordinates system, creates classes of CaseFaceOrientation that belong to the same mega-tile (all the orientations that you can reach in a mega-tile when you enter it with a particular CFO). I then linked all the classes in an adjacency matrix to define the "mega-space" or "mega-classes" that has to be explored. Generates images for convenience in the hopes that it would give me some visual insights of how to prove the rolling.


Classes relationships for the Cube rolling on checkerboard (matrix: completely filled). You can see all CFO classes are related.

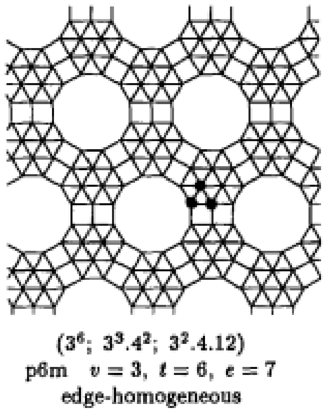

Classes relationships for the Tetrahedron on triangles (matrix: identity). You can see none of the CFO classes are related.

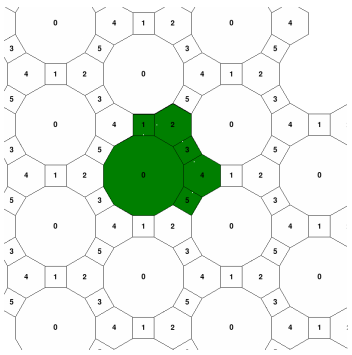

Classes relationships for the Square Pyramid on one of its net-tessellation (matrix: a big class followed by smaller ones). You can see one starting face has may relations, while others are stuck.

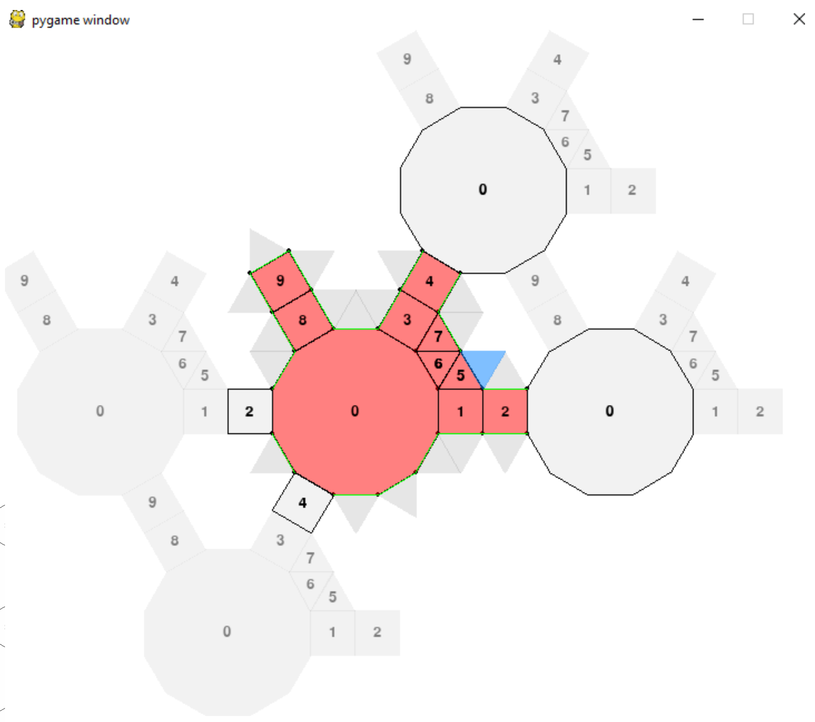

Classes relationships for the Octahedron on triangles. (matrix: two squares) You can see there are two separate classes, as we could have guessed from my first rolling visualisation that had bands of red and pink for the octahedron.

Big class relationships are a desirable property for Tiling/Polyhedron pairs to be used in face-matching puzzle games.

9.4 tiling_drawer.py, ScreenspaceRoller.py (Barbados jun. 2021)







9.4.1 tiling_drawer.py

First I created tiling_drawer.py, an evolution of NetDrawer.py, that allows to choose which shape to draw, to be able to convert the tilings we saw in the catalog to useable format. I wrote tiling_visualisation.py to turn that data into the images that we used during in the table.
The format describes the same "super-tile" that can be used to isohedrally tile the space as the Net-Tessellation use, as wel as the Coordinate system.

Based on feedback, I also updated the format I save my tiling as, making it easier to manipulate.

Instead of looking like this where the edge identifier was added as a multiple of the net size to the neighbour face:

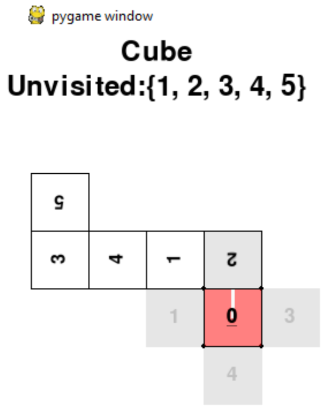
```
p=6
nets["cube_net_tessellation"] = {
    0: [1, 2, 4, 3],
    1: [0, 2 + p, 5, 5 + p],
    2: [0, 5 + p, 1 - p, 3 + p],
    3: [0, 4 + p, 4 + 2 * p, 2 - p],
    4: [0, 3 - 2 * p, 3 - p, 4+p],
    5: [5+p, 2 - p, 1 - p, 1]
}
```

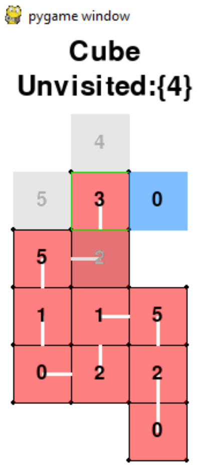
It now contains 2-tuples with the neighbour face, and the edge identifier separated.

```
biisogonal_tilings['(3x4^2x6;3x6x3x6)1'] = \
{   0: [(3, 0), (1, 0), (2, 0), (3, 1), (2, 2), (1, 3)],
    1: [(0, 0), (4, 0), (0, 3)],
    2: [(0, 0), (0, 2), (4, 5)],
    3: [(0, 0), (4, 4), (0, 1), (4, 0)],
    4: [(1, 0), (3, 0), (2, 5), (3, 4)]}
```

9.4.2 UnlimitedNetDrawer.py, tool to help visualise rolls by hand

Then I created UnlimitedNetDrawer.py, an evolution of NetDrawer.py that allows to draw in only one direction but does not have a size limit (hence "Unlimited"), to help people look for hamiltonian paths by hand by rolling a shape. I also loaded it with all regular-faced polyhedrons I could find and convert.





9.4.3 ScreenSpaceRoller.py, new improved rolling visualisation

Based on the new tilings found for the Barbados workshop, I modified my exploration code to accept separate tilings. It was the version with the very heavy visuals (faces and orientation ticks). Then I was encouraged to update my exploration code based on several hints by Aaron:

- Make it easier to disable the visualisation, or pass different visualisations.
- Define the search area as two areas: a center area where you look for results, and a bigger area that takes care of the bad edges. So you can output a binary result without human input: was the center fully explored or not?
- This lead to me realising that the reason my previous attempts were not covering the whole space sometimes is because I was implicitly defining my search space as one Mega-Tile only (the Net-Tessellation was the size of the search list) and that's why I had trouble seeing the whole image. Switching to arbitrary tessellation meant that this restriction looked more and more difficult to justify to people, so I should instead switch to recording the exploration of all the coordinates on the screen.
- Use breadth-first to focus the search to the center of the screen, and thus spend more time doing useful searching. (Use a queue instead of a stack.)
- I could find symmetries in the rolling polyhedron by looking for graph isomorphism. I first tried by looking at the Adjacency matrix by switching the order of faces, but the results were unreliable. Aaron suggested that I modify the dual graph of the polyhedron I roll by adding something like antennaes to the faces and orientation I want to compare, then use an external tool to check for isomorphism. Using symmetries data lead to huge speed improvements.
- Stefan suggested that I color the explored space based on how many classes of FO pairs reached that cell, rather than use transparency to overlay the number of times the cell had been visited previously as I was doing. In addition to looking better, this also happened to speed up the drawing process because drawing alpha transparency is slower than generating a new solid color.

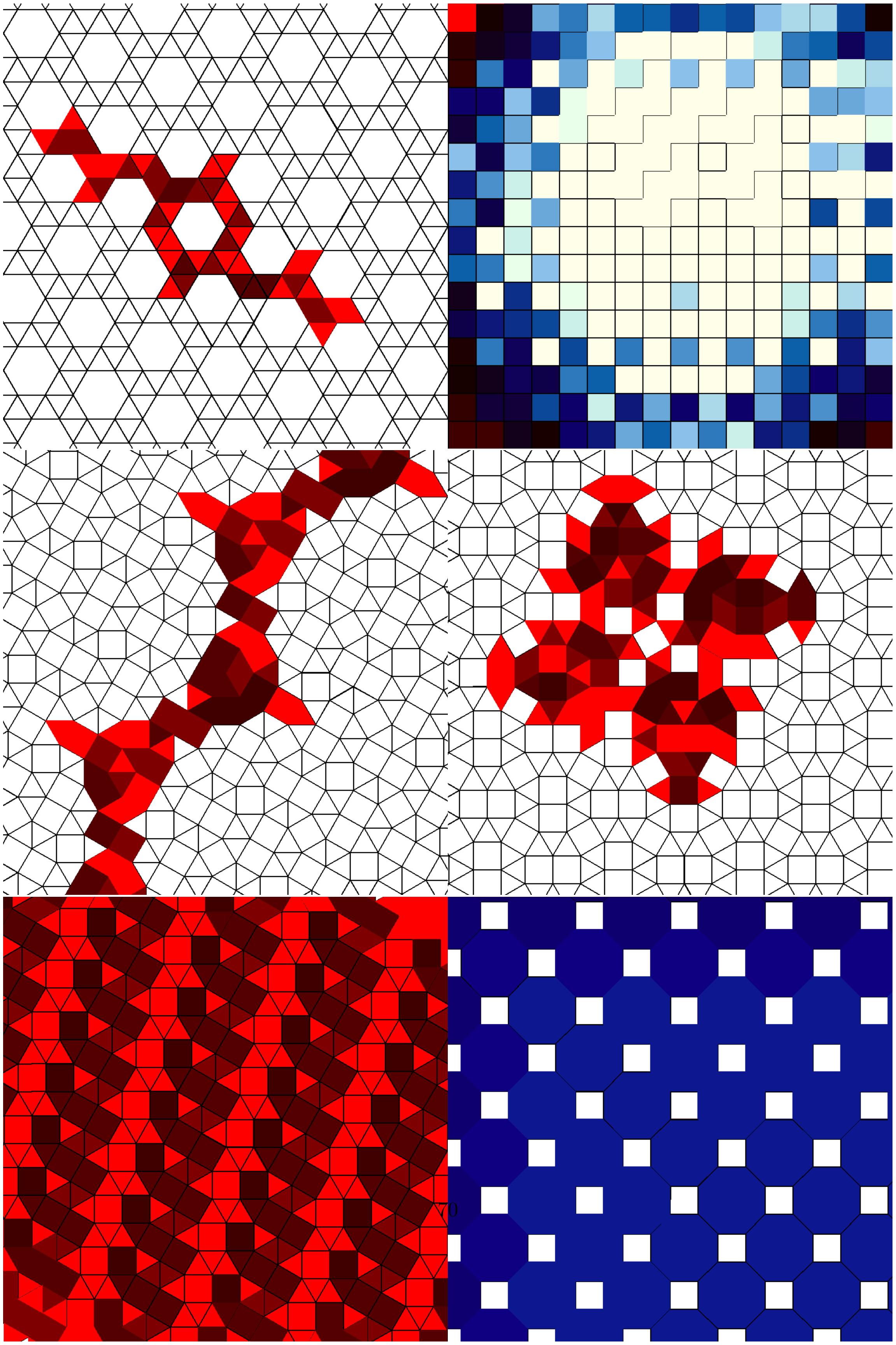


Figure 9.3: Screenspace roller's output

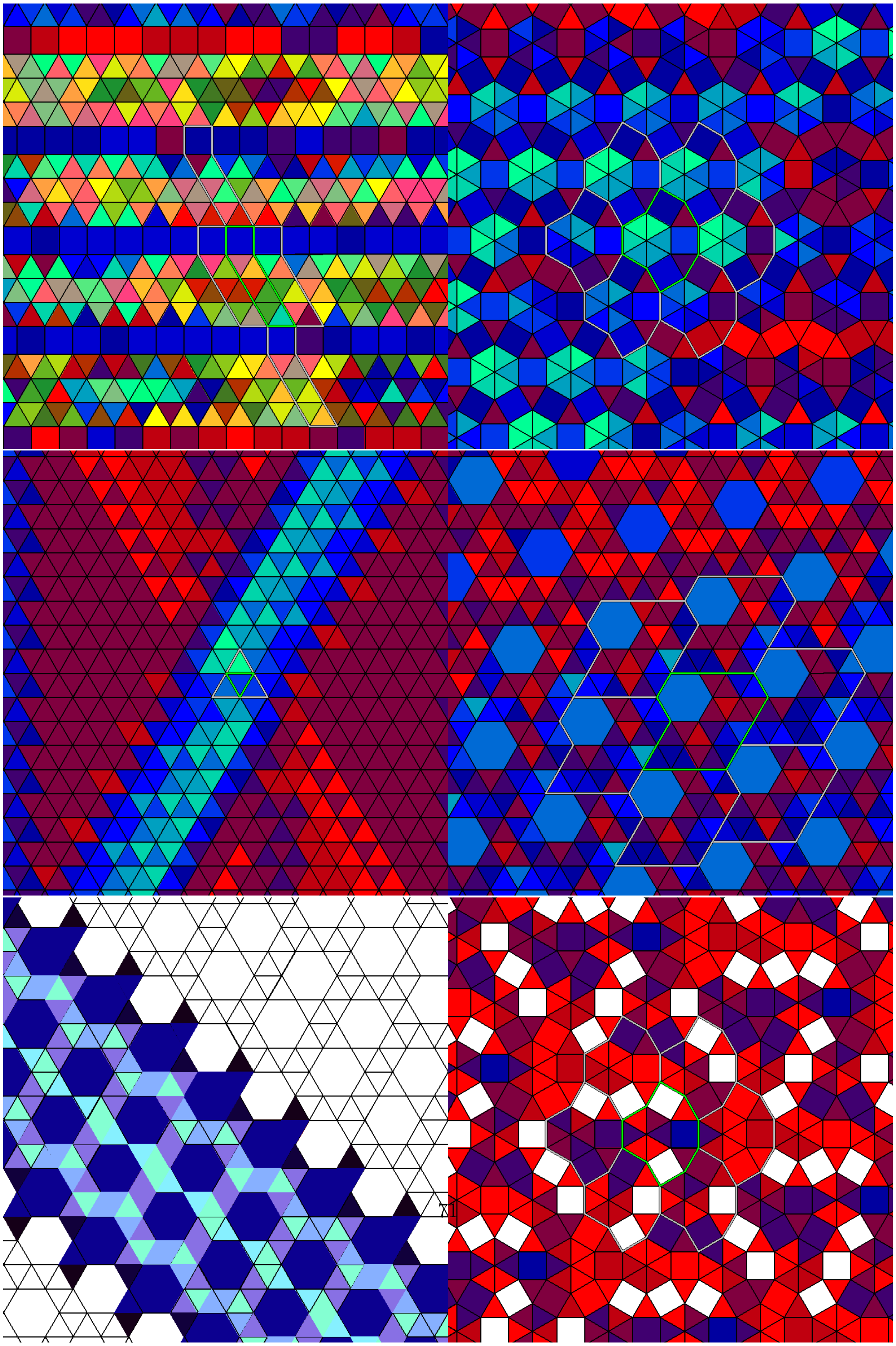
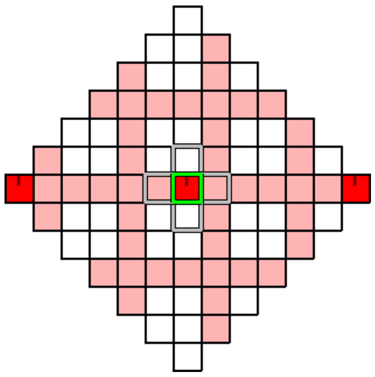
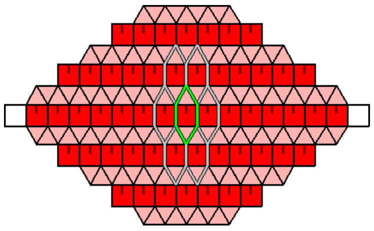
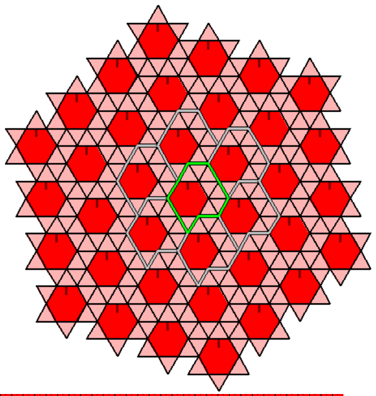
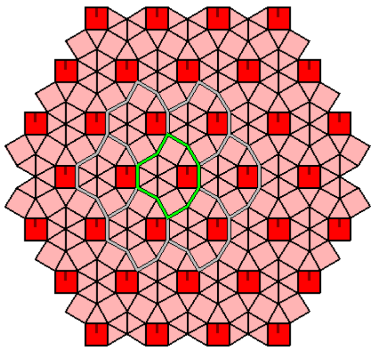
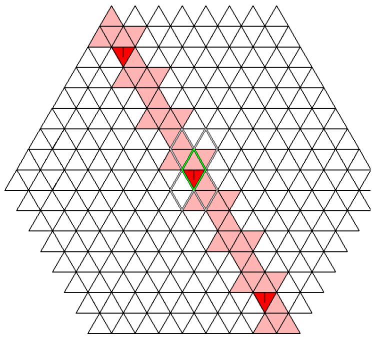
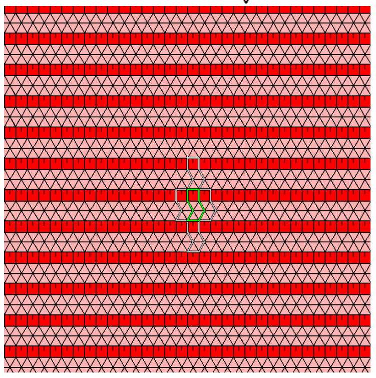


Figure 9.4: Screenspace roller's output with a different coloration rule

9.5 N-space roller (july 2021)

In order to generate a "proof" that a roller rolls the whole space, screenspaceroller was outfitted with an option to limit the search space to N-tiles around the center, to fill an N/2-size area around the center that could be used as representative area. The visualisation was again changed to highlight symmetrical $< c, f, o >$ states (with a notch for their orientation), but only of the starting state rather than the whole starting supertile. However, the calculated N was based on number of states, but was multiplying the number of supertiles which contained several states, meaning most representations were too big to read. This would lead to both the idea of limiting the presentation space with symmetry vectors and using the superposition of states in a supertile as our factor for determining N. Highlighting the supertile and its neighbours was also a sign of shifting the way to consider it.

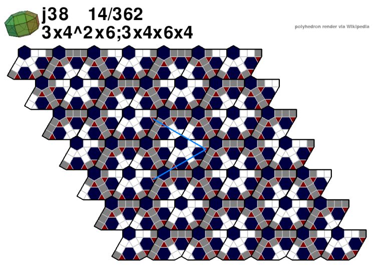
9.6 RollingProof.py (july-august 2021)

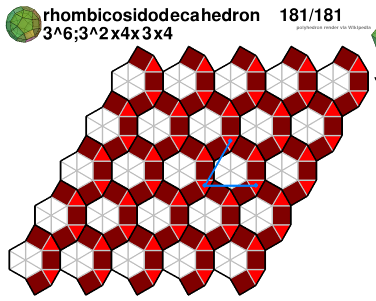
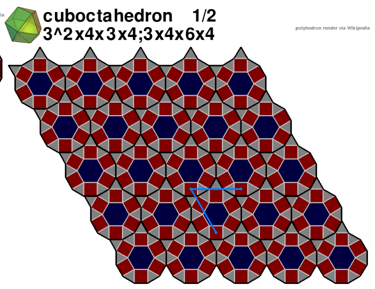
The need to write a formally defined proof to test for the rolling property lead to a lot of shifting concepts. Exploring a ”big enough area around a smaller representative area” was not good enough, as writing a proof with that concept was not panning out for determining the lower bound of that ”big enough area”.

A new proof, using the IsohedralTileRollExplore’s CFO classes as a base, was written. Instead of using Conway coordinates system, it uses regular cartesian coordinates for parallelogon supertiles, which made the determination and usage of symmetry vectors a lot easier. Thanksfully in the tilings derived from the catalog, all but one tessellation were already parallelogon. The missing one, the triangular tiling, could easily be redefined into one. The tessellation polyhedrons were not ported over.

The search was also separated from the image output. The search now outputs a pickled dictionary which is sent to another script to draw data from. The output image is also now resized before being exported.

Interacting with puzzle makers and thinking about the practical uses of the research also lead to a shift in which information to display, focusing practical informations like face-complete tiles and stability over visual flair.




rhombicosidodecahedron 181/181
3^6;3^2x4x3x4
polyhedron render via Wikipedia


Chapter 10

Abandoned directions discussions

10.1 Abandoned proof type

Another way to prove that the roller covers the space would be to take the Representative Area of size N and explore around it until all tiles are covered by a connected path without making use of symmetry information. However the number of tiles to explore to find all connected covering paths is hard to define and could be as big as $O(N^2)$.

Definition 55. $\sim$ meaning there is a valid roll between the two positions/states/tiles.

Fact 8. Two tiles t_0, t_k with two states sp_0, sp_k are connected if there exist a path
Two positions $< t_0, sp_0 >, < t_k, sp_k >$

$$< t_0, sp_0 > \sim < t_1, sp_1 > \sim \dots < t_{k-1}, sp_{k-1} > \sim < t_k, sp_k > \tag{10.1}$$

The upper bound on path length seems to be close to N^2. The issue is that we can't remove repeated states from the path because we are trying to reach a specific position in the non-periodic rolling graph.
However, we know that an unique path can stray by at most N tiles from either position, before using symmetrical loops to fill the offset. Soin term of distances, N tiles around the representative area might be enough?

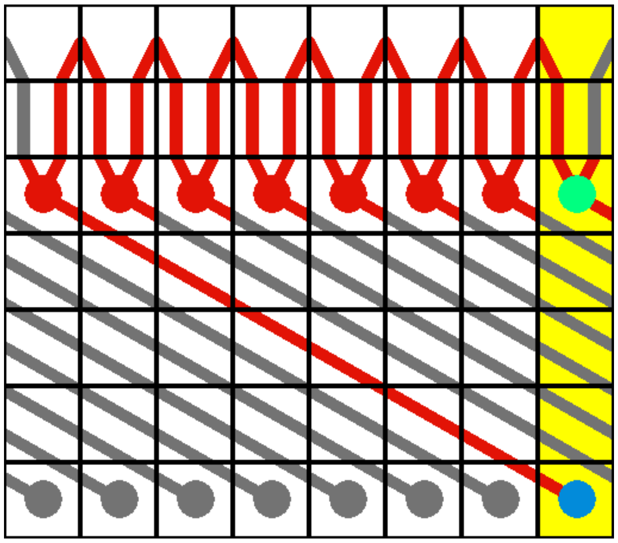


Figure 10.1: Graph structure with a long external path that decides of the connectivity of two positions.
Path length is at most $(N/2)^2 + N/2$
Is this the longest possible external path type?

Fact 9. A cycle on the cartesian plane coordinates passing trough t_0 is defined by

$$t_0 \sim\sim t_1 \sim\sim ...t_{k-1} \sim\sim t_k = t_0 \tag{10.2}$$

Fact 10. A cycle on the periodic rolling graph passing trough sp_0 is defined by

$$sp_0 \sim\sim sp_1 \sim\sim ...sp_{k-1} \sim\sim sp_k = sp_0 \tag{10.3}$$

A cycle over the two might mean they must synch up, so $|N| * |S|$ should be enough of a search size. This is a lot larger than the size I use in practice to get results.

10.2 Abandoned Conway Tile Coordinate system

Remark. In the end, this form of non-cartesian coordinates system was abandoned in favor of cartesian coordinates with parallelogons.

Definition 56 (Coordinate system)**.** A coordinate system uses n numbers, or coordinates, to uniquely identify an element in a $n-space$. The $n-space$ can be described by a canonical $n-base$ of n linearly independent elements. The coordinates correspond to a multiplicative factor of each of the element in the base.

Definition 57 (Conway Criterion)**.** The Conway criterion identifies a family of tessellation tiles that tile the plane using translation and 180-degree rotations. The tile's borders must have

- Two matching sides that are a translation
- Several sides that match themselves as centrosymmetries

Remark. We selected the conway criterion because all tessellation polyhedron's tilings found in [21] abide to the Conway Criterion.

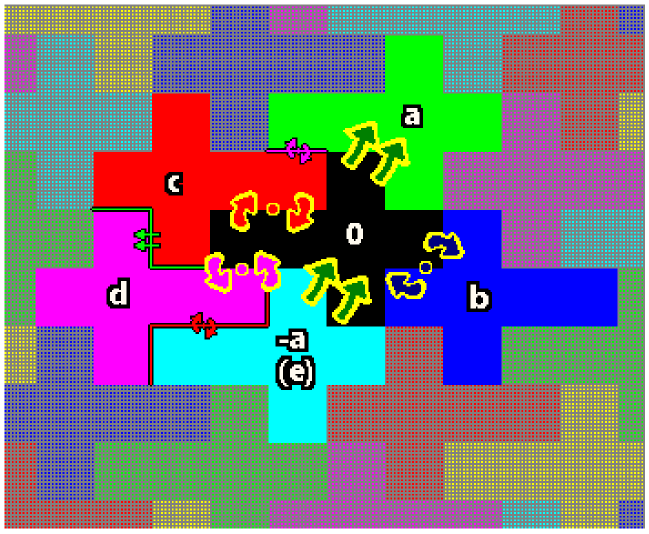
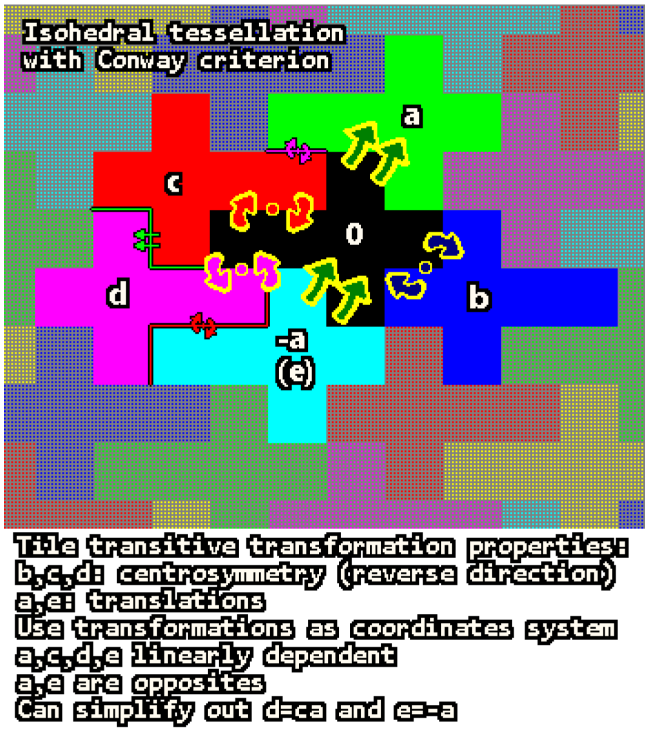


Figure 10.2: List of neighbours a,b,c,d,e of a tile.

We can see every Conway requirement as a neighbouring tile, and a tranformation. Each of those represent a direction. By calling one transformation we move from one tile to another. By combining the directions we can move to, we can create a base, and thus a coordinate system.

Theorem 12. For every tessellation that abides to the Conway Criterion, using the requirements transformations as base, and using neighbouring Linear Algebra to simplify it, starting from a tessellation tile, we can create a canonical Base of Neighbours, and thus a Coordinate System for the Tile in the Tessellation.

- The pair of translation tiles can serve as one axis (one direction is positive, the other negative).
- The rest of the neighbours are central symmetries. Since this involves rotating 180°, if we follow this direction several times, we just turn back. To use it as an axis, we need to use a trick: applying that transformation twice nullifies it, but applying it after a translation moves us further. This means that everytime we add one on this axis, we reverse the sign on the increments of all the axes.
- Simple "Neighbour's neighbours" algebra show us that to get a coherent system, we must simplify out some relationships to get a canonical base.



Tile transitive transformation properties:
b,c,d: centrosymmetry (reverse direction)
a,e: translations
Use transformations as coordinates system
a,c,d,e linearly dependent
a,e are opposites
Can simplify out d=ca and e=-a

Figure 10.3: List of neighbours a,b,c,d,e of a tile, simplified into a 3-base [a;b;c]

Lemma 5. Every tile in the tessellation has an unique coordinate formed by the path of the transformations used to reach the tile.

Corollary 2. The unique coordinate formed by the path to reach a tile simplifies must always simplify to the same unique coordinate. Example: Figure 10.4 and Figure 10.5 simplify to the same coordinates.

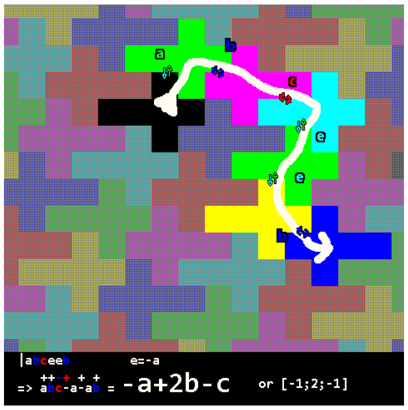


Figure 10.4: Long path [abceeb] simplified to -a+2b-c or [-1;2;-1]

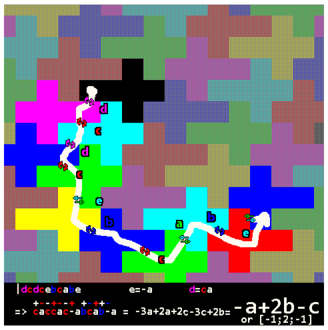


Figure 10.5: Longer path [dcdcebcabe] simplified to -a+2b-c or [-1;2;-1]

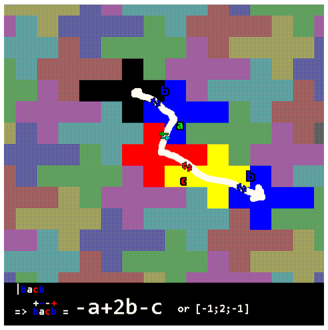


Figure 10.6: Shortest path [bacb] or -a+2b-c or [-1;2;-1]

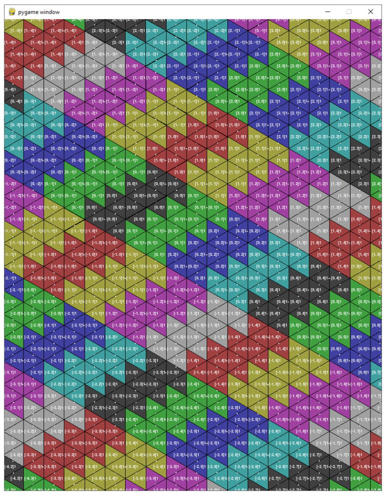
Corollary 3. The simplified coordinate of a tile is the shortest path from the origin tile to this tile. Example: Figure 10.6

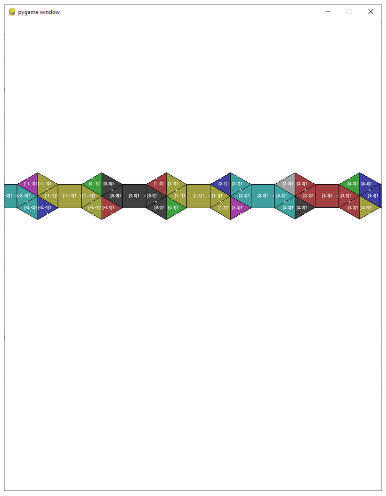
10.2.1 Manipulating the coordinates system

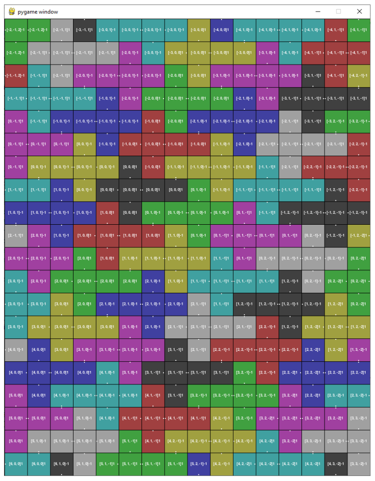
- Two centrosymmetries of opposite sign cancel eachother out. This means that two centrosymmetries applied successively will cancel eachother out, as applying a centrosymmetry inverts the sign of the next operations.
- Two translations of opposite sign cancel eachother out.
- You can insert a centrosymmetry of a translation anywhere, as long as the sign checks out, the order doesn't matter.
- This means as long as you know the sign, it is easy to re-order a sequence of symmetries to a sorted form. By simplifying the sorted form, you find the canonical form which gives the coordinates.
- Two sequences can be added up or substracted by respecting the sign of their last operation

Remark. Negative centrosymmetries don't exist as transformation, but they can show up as the sign of the operations gets negative. A negative centrosymmetry should be interpreted as the order in which it is applied (odd placement in the list of centrosymmetries). This means that a negative centrosymmetry doesn't exist without a positive centrosymmetry balancing it out.

Remark. Mirror symmetries should also be taken in account next. Mirror symmetries are like centrosymmetries but they don't invert the sign of the translations?







The coordinates system used to color net-tessellations of j17, j1, cube

Chapter 11

Result tables

11.1 Plane rollers table

Legend

Roll column (reachable areas and face-completeness)

- Blue lines: repeating states symmetry vectors
- Grey cells: reached areas
- Brown cells: areas reached with all compatible faces from the polyhedron
- Red cells: areas reached with all compatible faces from the polyhedron in all orientations
- White cells (non-roller): not reached
- Black cells (quasi-roller): areas incompatible with every face of the polyhedron

Stability column (for unstable rollers, tiles that always generate a roller)

- Grey cells: stable cells: the structure of the rolling graph does not change depending on the face/orientation that starts on this area
- White cells: unstable cells, might lead to the roller getting stuck in an bounded area or band

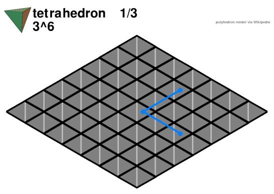
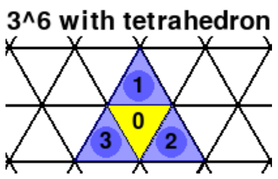
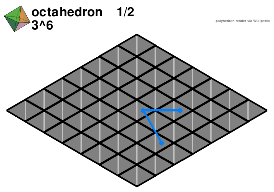
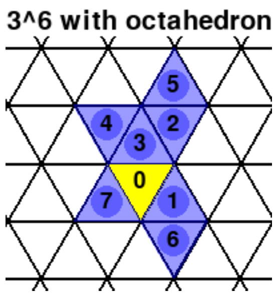
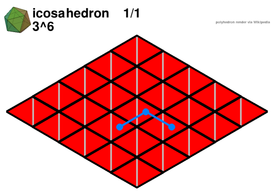
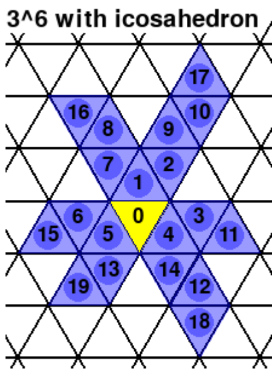
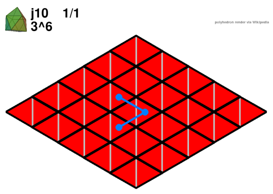
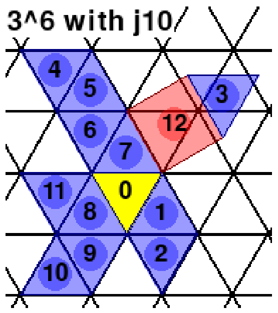
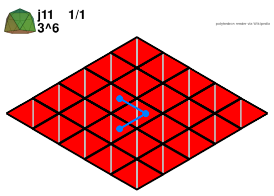
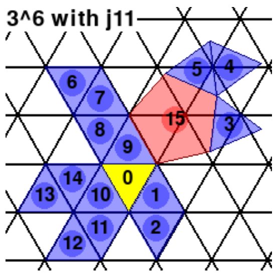
Stable rollers would have the entire area greyed out.

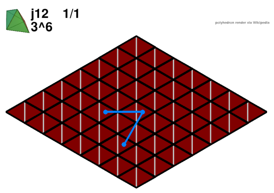
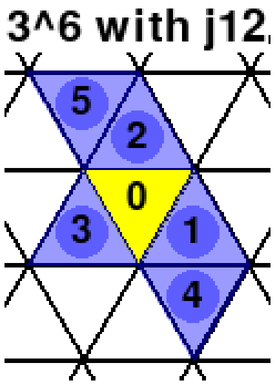
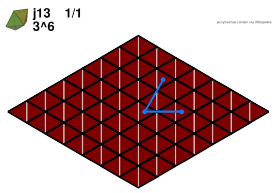
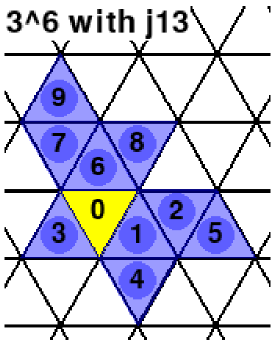
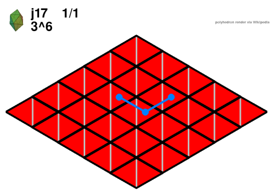
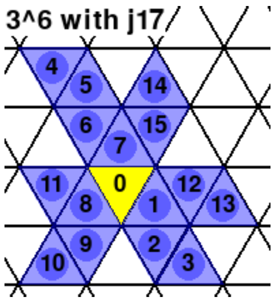
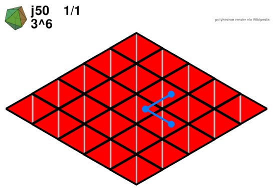
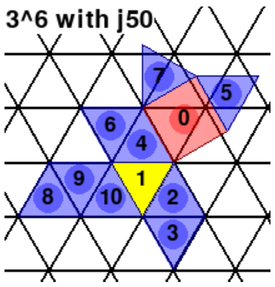
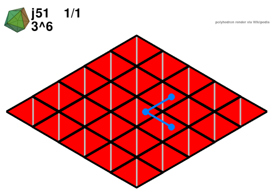
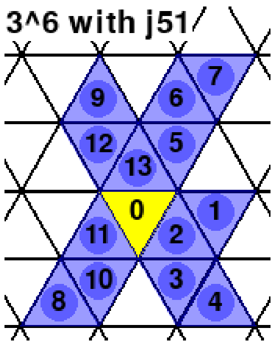
Faces column (net of the polyhedron, example of starting position)

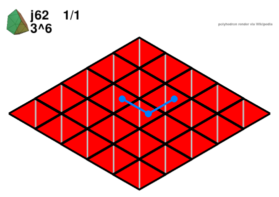
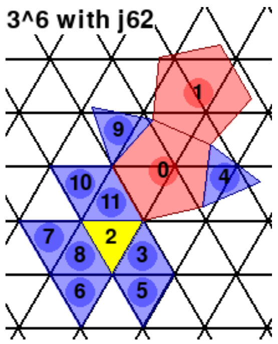
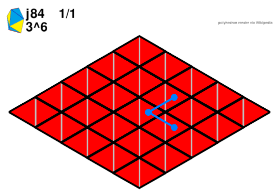
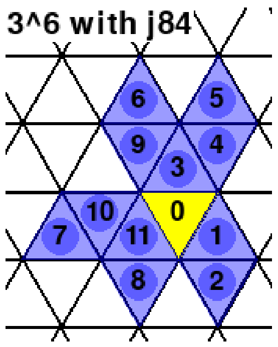
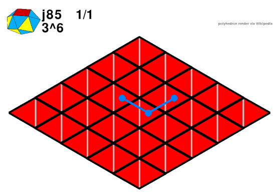
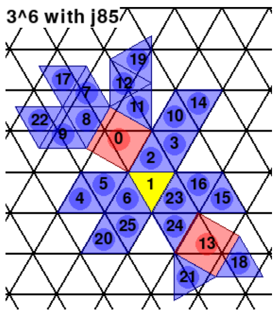
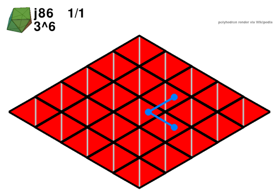
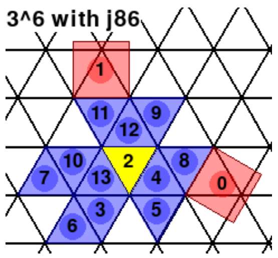
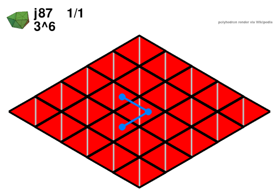
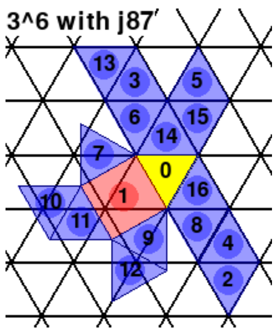
- Yellow face: example of starting face and orientation in the net
- Red face: unused face in the rolling graph. Can be either incompatible, or not be part of the connected component that rolls the plane.

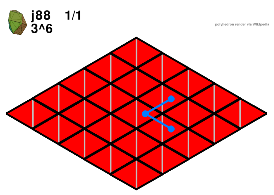
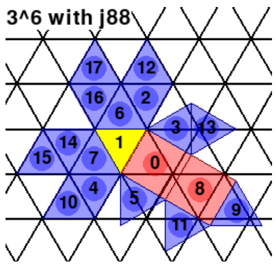
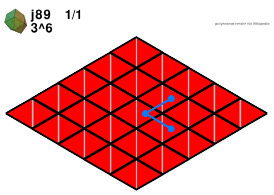
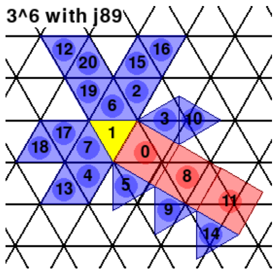
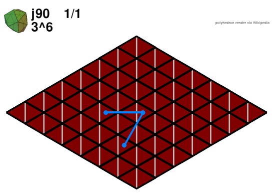
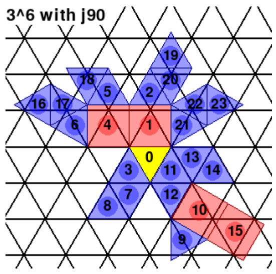
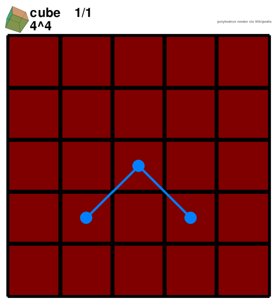
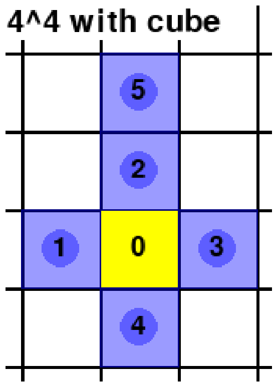
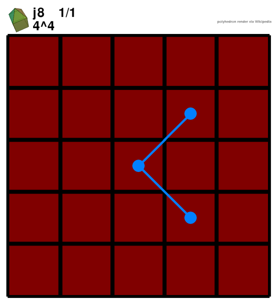
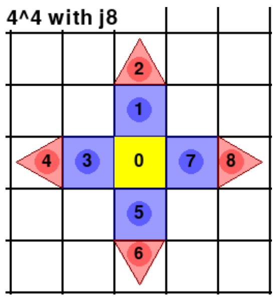
Tables start next page.

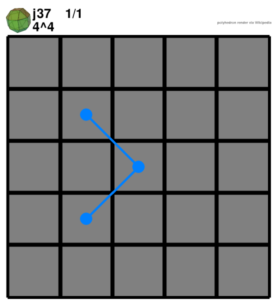
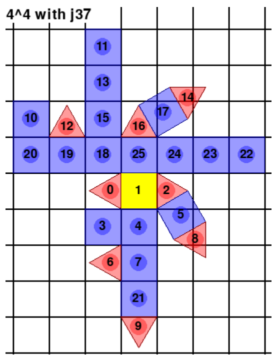
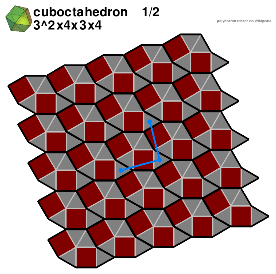
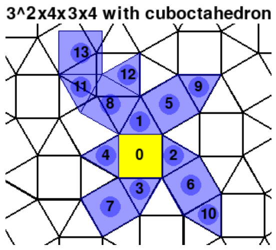
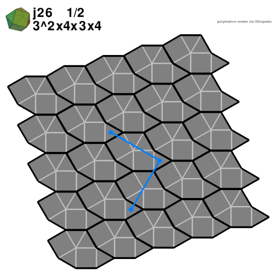
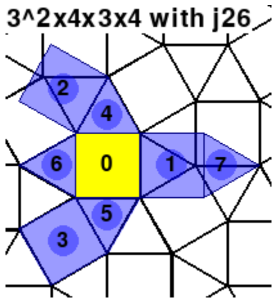
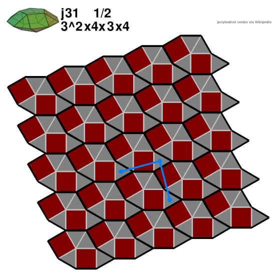
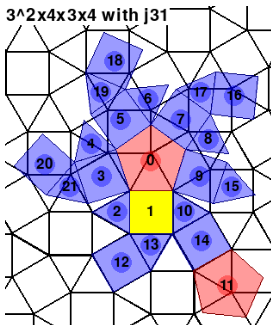
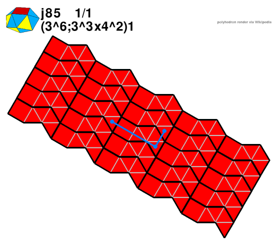
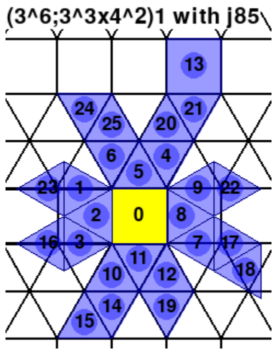
11.1.1 Stable Plane rollers table

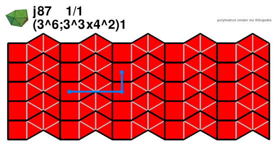
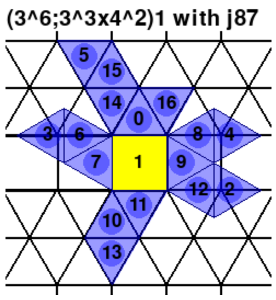
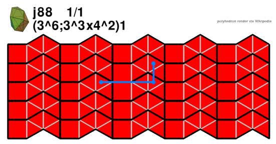
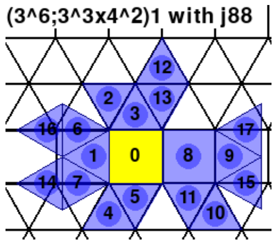
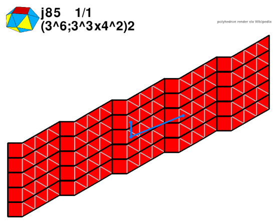
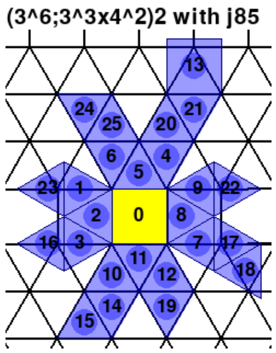
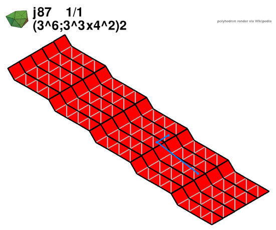
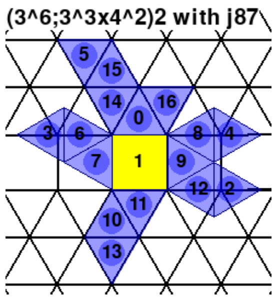
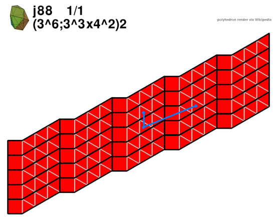
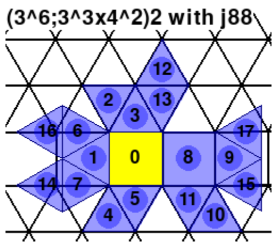
Pair	Roll	Stability	Faces
tetrahedron 3^6 back		All tiles	
octahedron 3^6 back		All tiles	
icosahedron 3^6 back		All tiles	 3^6 with icosahedron 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
j10 3^6 back		All tiles	
j11 3^6 back		All tiles	

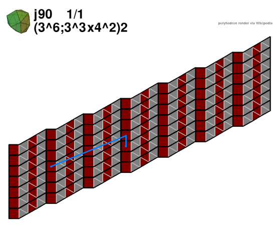
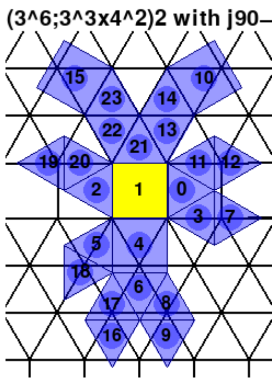
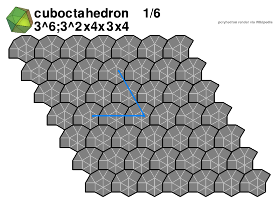
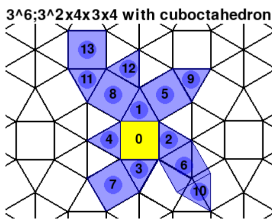
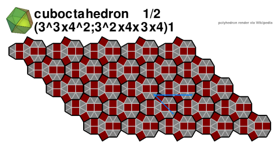
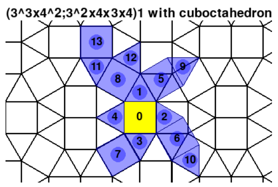
j12 3^6 back		All tiles	
j13 3^6 back		All tiles	
j17 3^6 back		All tiles	
j50 3^6 back		All tiles	
j51 3^6 back		All tiles	

j62 3^6 back		All tiles	
j84 3^6 back		All tiles	
j85 3^6 back		All tiles	
j86 3^6 back		All tiles	
j87 3^6 back		All tiles	

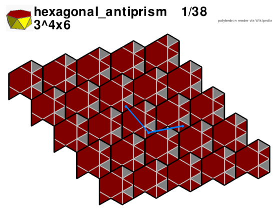
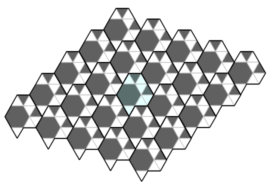
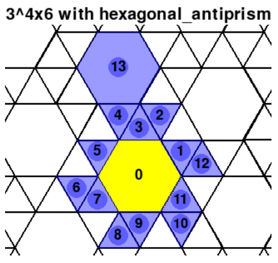
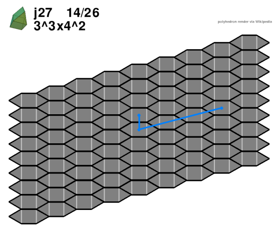
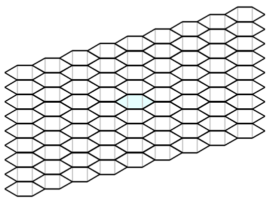
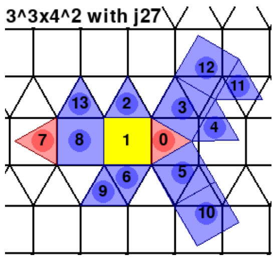
j88 3^6 back		All tiles	
j89 3^6 back		All tiles	
j90 3^6 back		All tiles	
cube 4^4 back		All tiles	
j8 4^4 back		All tiles	

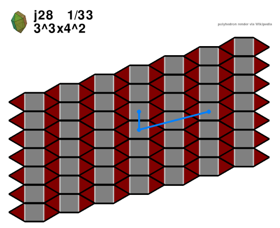
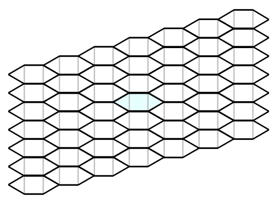
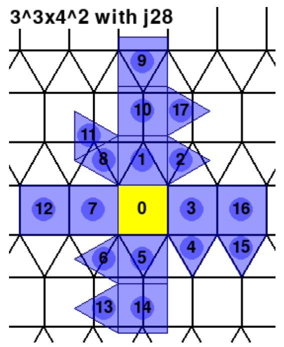
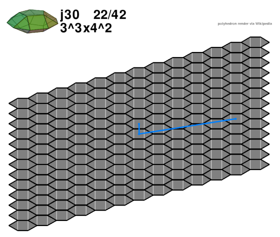
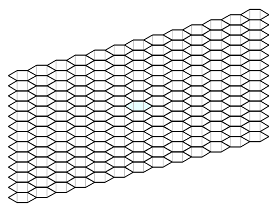
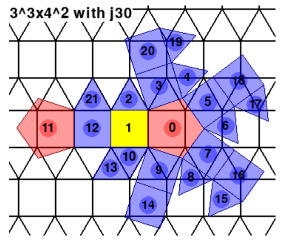
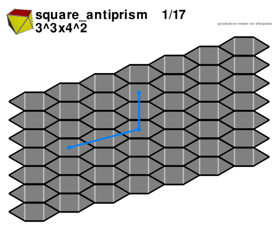
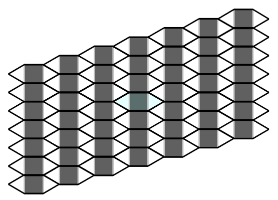
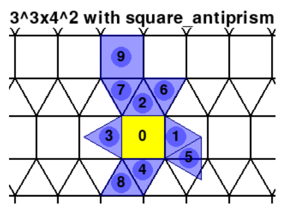
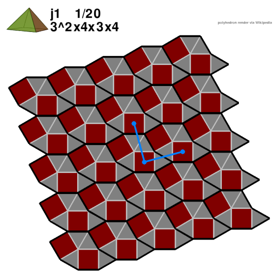
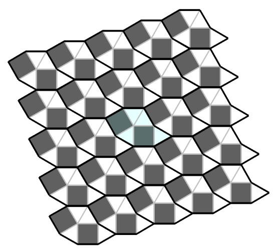
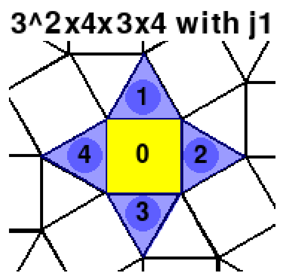
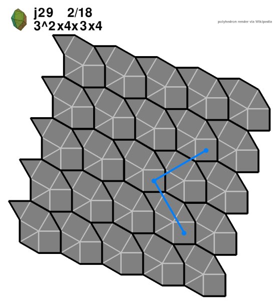
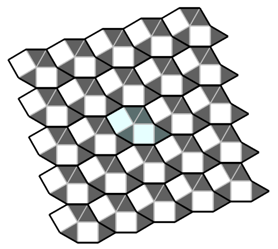
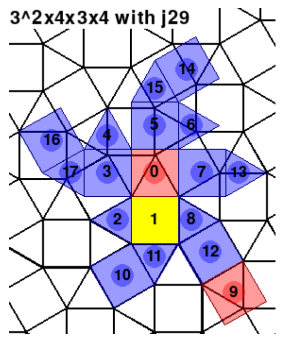
j37 4^4 back		All tiles	
cuboctahedron $3^2x4x3x4$ back		All tiles	
j26 $3^2x4x3x4$ back		All tiles	
j31 $3^2x4x3x4$ back		All tiles	
j85 $(3^6;3^3x4^2)1$ back		All tiles	

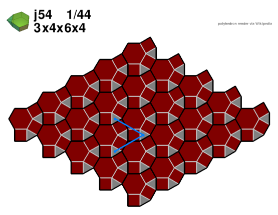
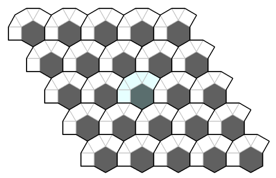
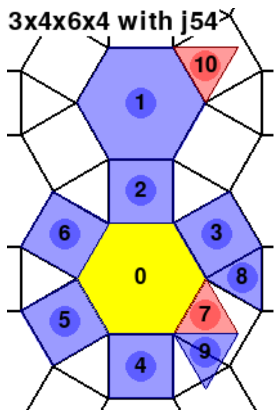
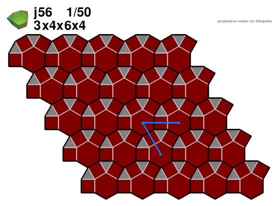
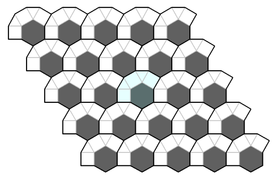
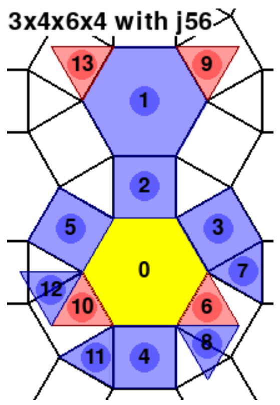
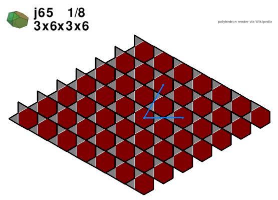
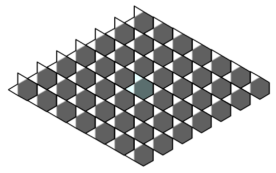
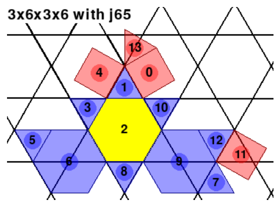
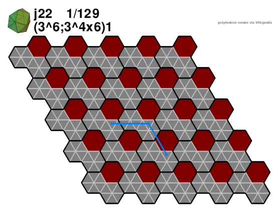
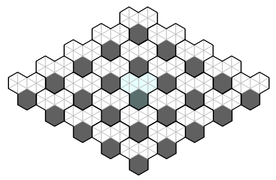
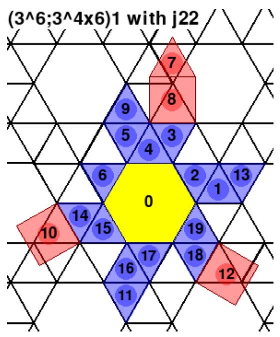
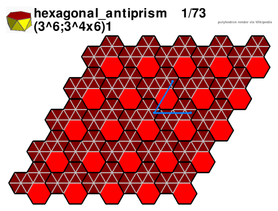
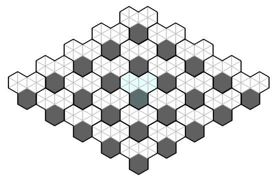
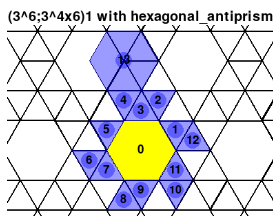
j87 $(3^6; 3^3x4^2)1$ back		All tiles	
j88 $(3^6; 3^3x4^2)1$ back		All tiles	
j85 $(3^6; 3^3x4^2)2$ back		All tiles	
j87 $(3^6; 3^3x4^2)2$ back		All tiles	
j88 $(3^6; 3^3x4^2)2$ back		All tiles	

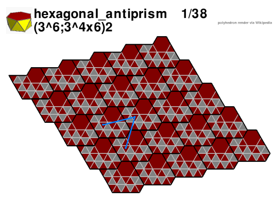
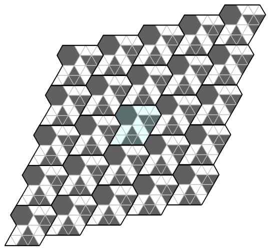
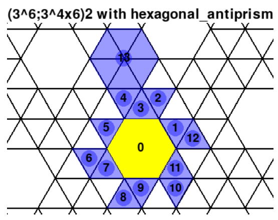
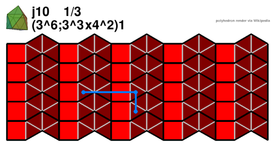
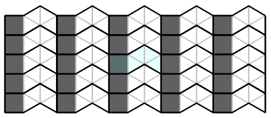
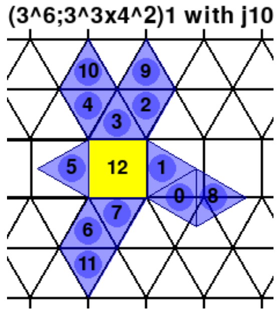
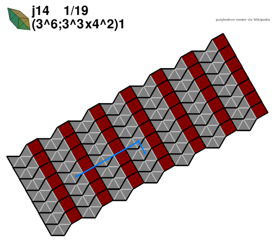
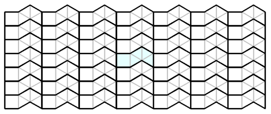
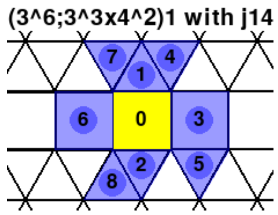
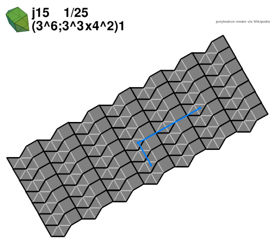
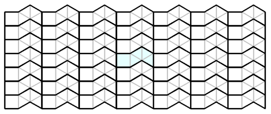
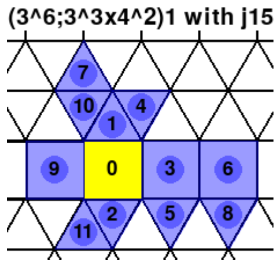
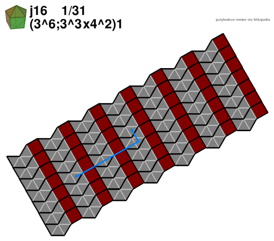
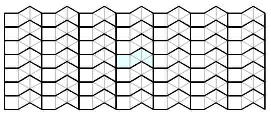
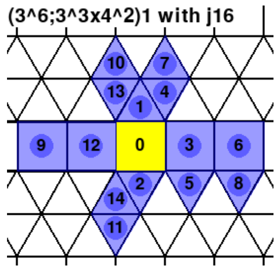
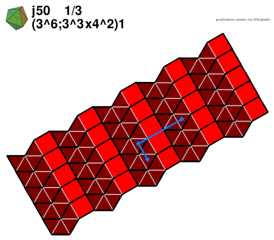
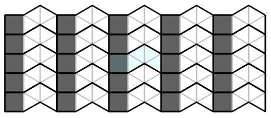
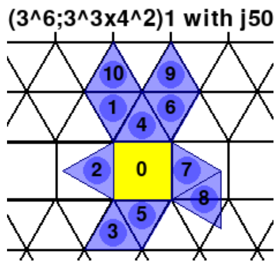
j90 $(3^6; 3^3x4^2)2$ back		All tiles	
cuboctahedron $3^6; 3^2x4x3x4$ back		All tiles	
cuboctahedron $(3^3x4^2; 3^2x4x3x4)1$ back		All tiles	

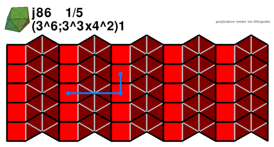
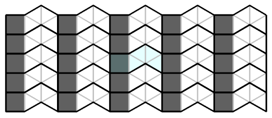
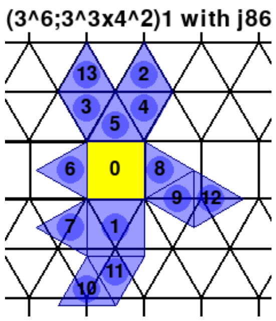
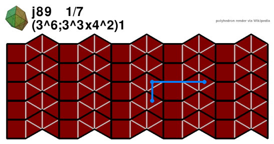
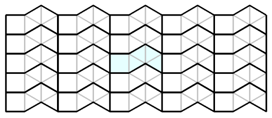
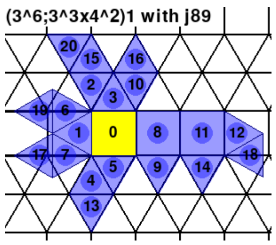
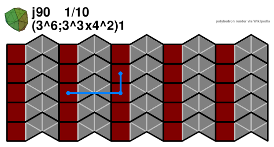
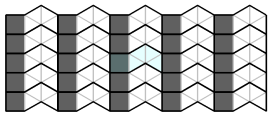
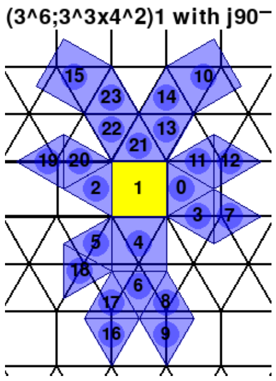
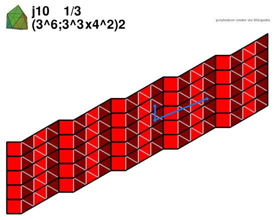
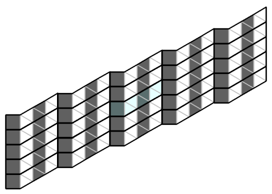
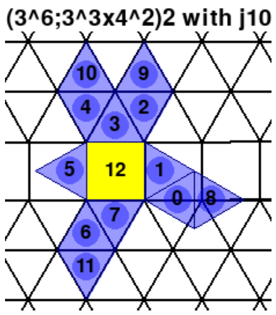
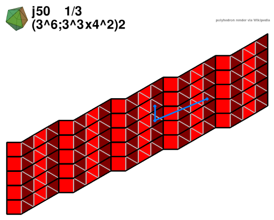
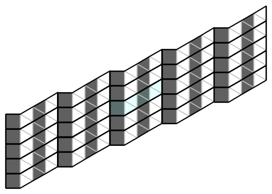
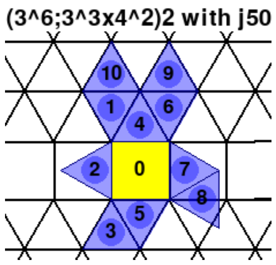
11.1.2 Unstable Plane rollers table

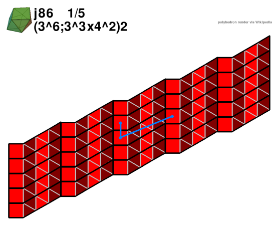
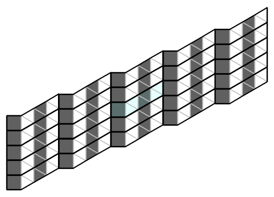
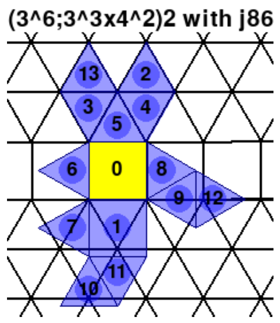
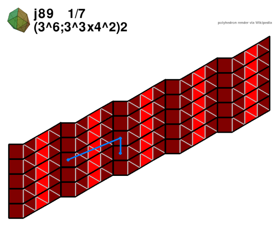
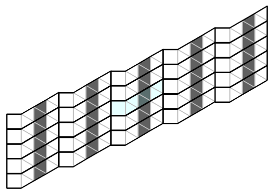
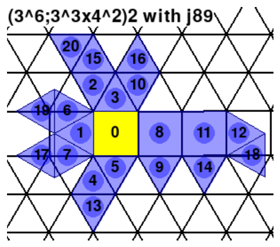
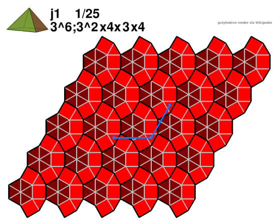
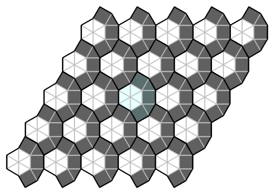
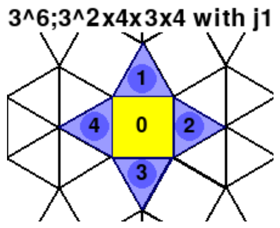
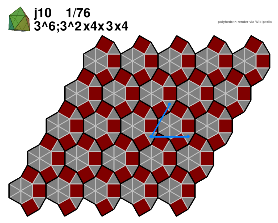
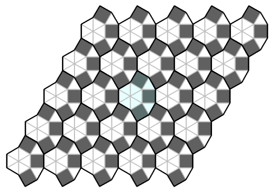
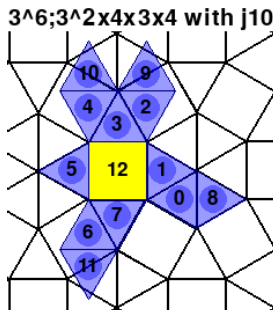
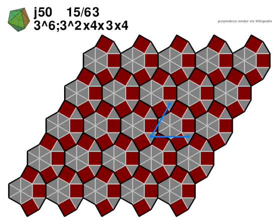
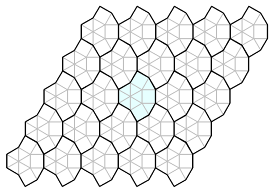
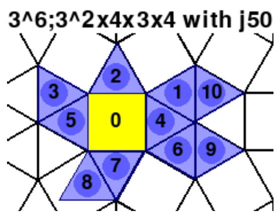
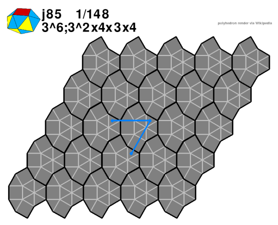
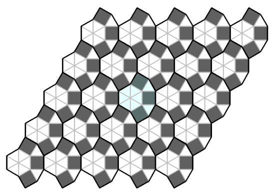
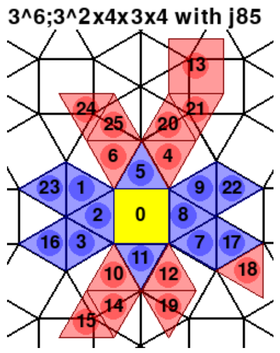
Pair	Roll	Stability	Faces
hexagonal antiprism 3^4x6 back			
j27 3^3x4^2 back			

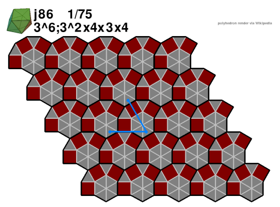
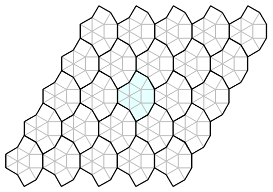
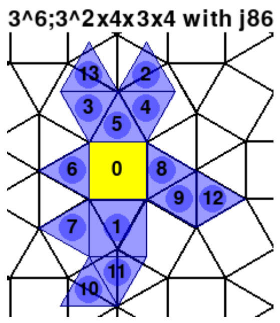
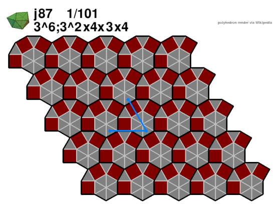
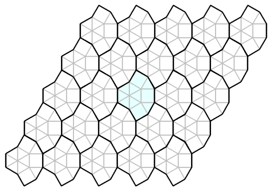
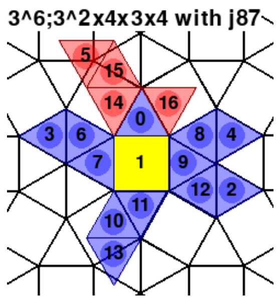
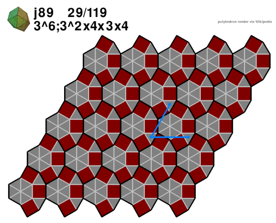
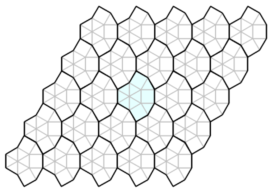
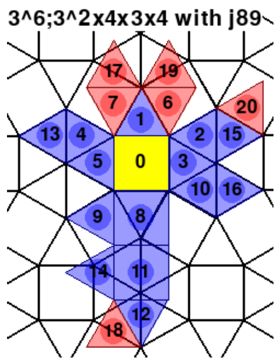
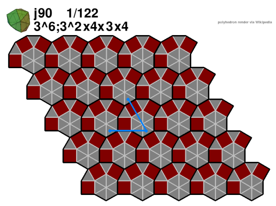
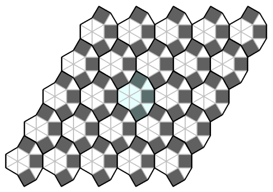
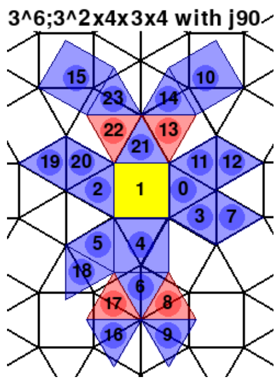
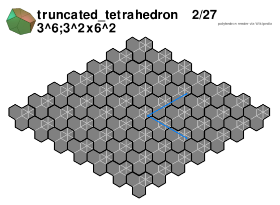
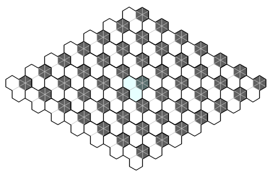
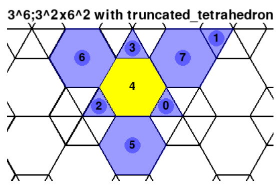
j28 3^3x4^2 back			
j30 3^3x4^2 back			
square antiprism 3^3x4^2 back			
j1 $3^2x4x3x4$ back			
j29 $3^2x4x3x4$ back			

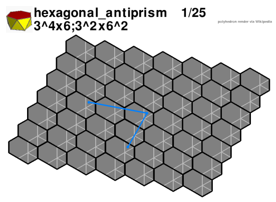
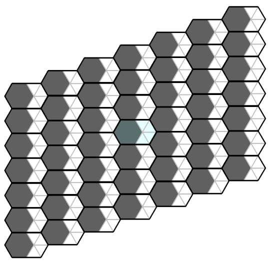
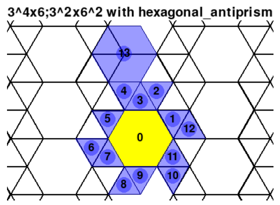
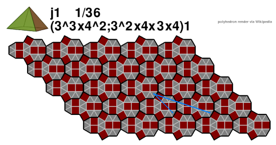
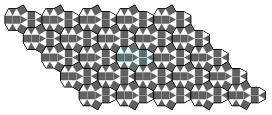
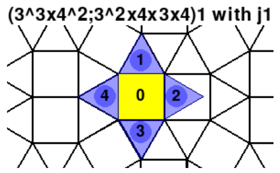
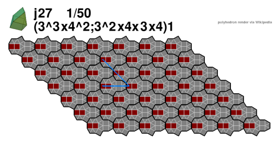
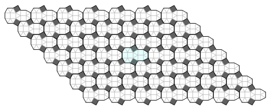
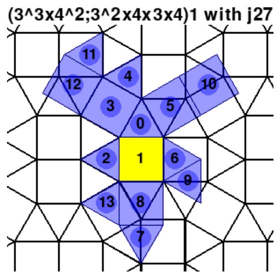
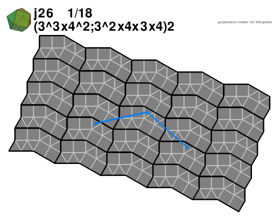
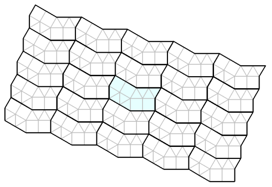
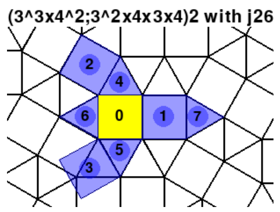
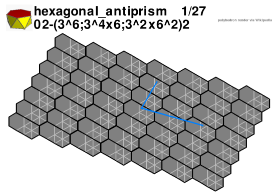
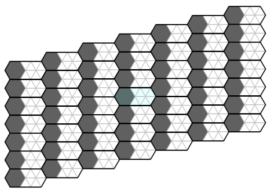
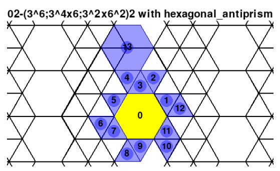
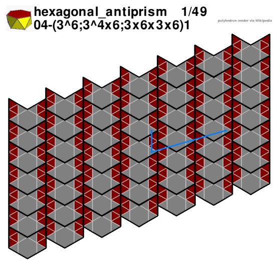
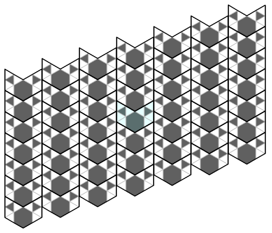
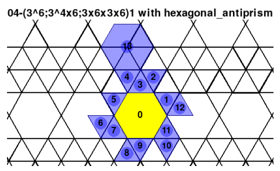
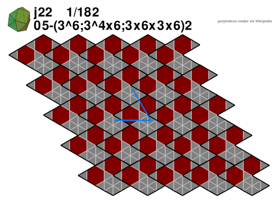
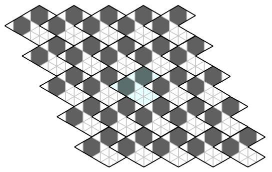
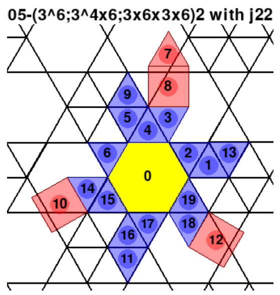
j54 $3x4x6x4$ back			
j56 $3x4x6x4$ back			
j65 $3x6x3x6$ back			
j22 $(3^6; 3^4x6)1$ back			
hexagonal antiprism $(3^6; 3^4x6)1$ back			

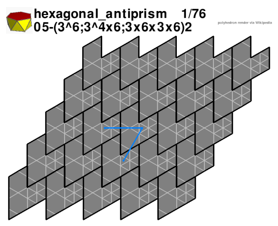
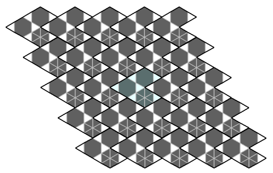
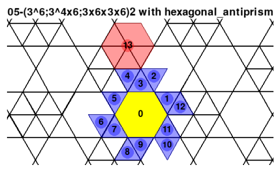
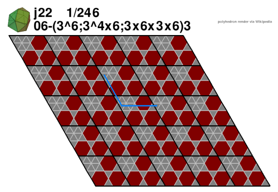
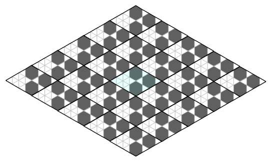
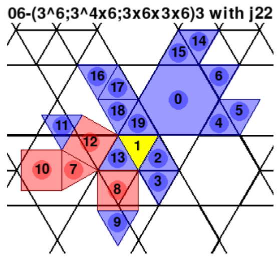
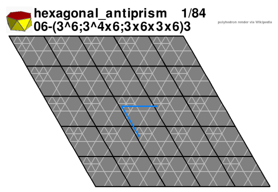
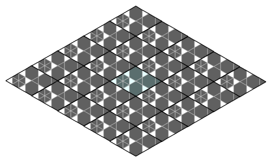
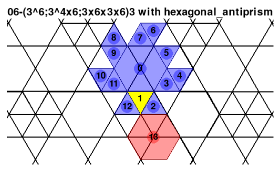
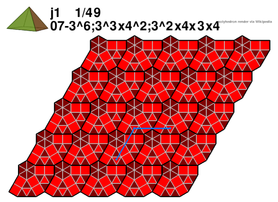
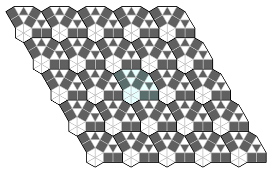
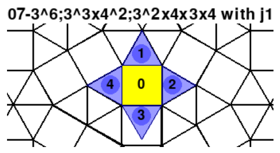
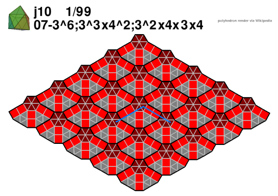
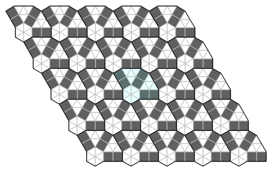
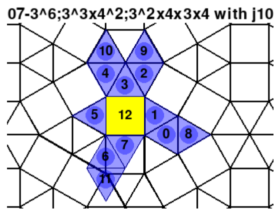
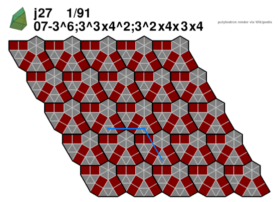
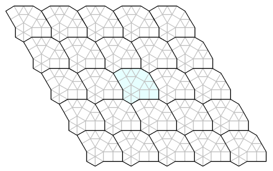
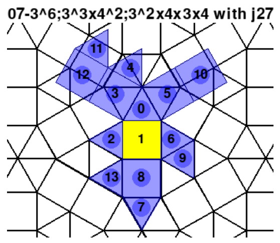
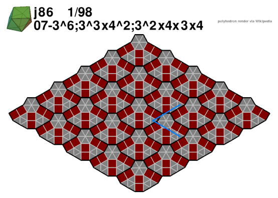
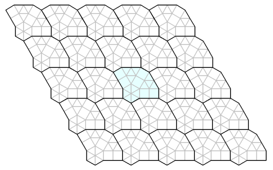
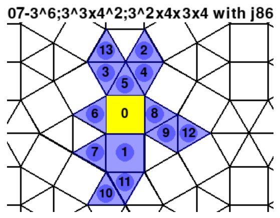
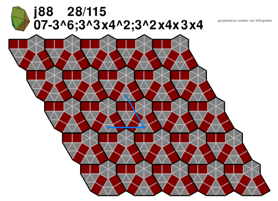
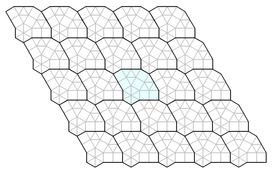
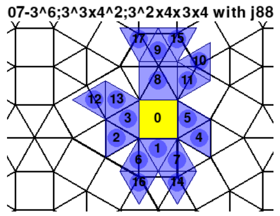
hexagonal antiprism $(3^6;3^4x6)2$ back			
j10 $(3^6;3^3x4^2)1$ back			
j14 $(3^6;3^3x4^2)1$ back			
j15 $(3^6;3^3x4^2)1$ back			
j16 $(3^6;3^3x4^2)1$ back			
j50 $(3^6;3^3x4^2)1$ back			

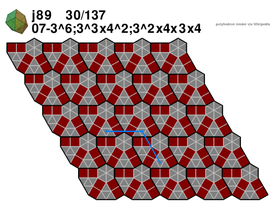
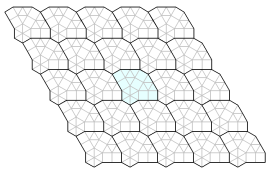
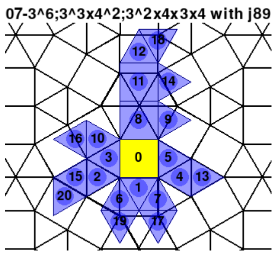
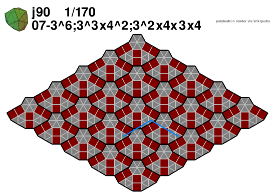
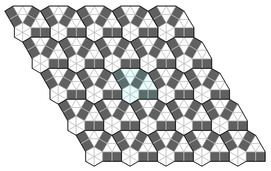
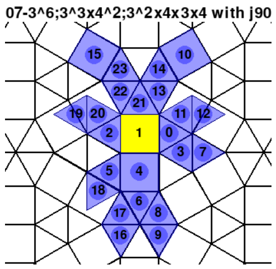
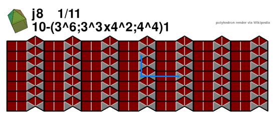
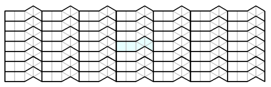
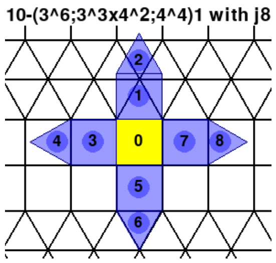
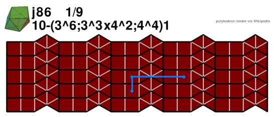
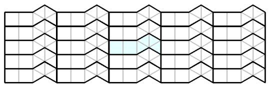
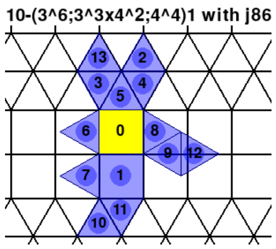
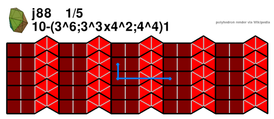
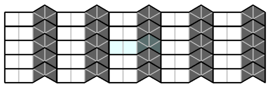
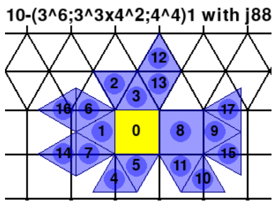
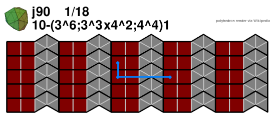
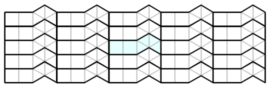
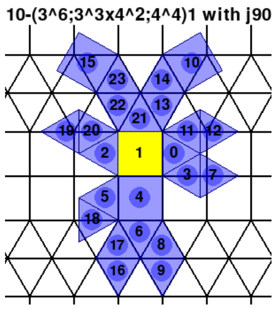
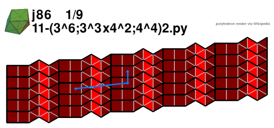
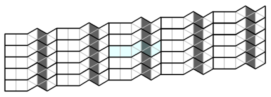
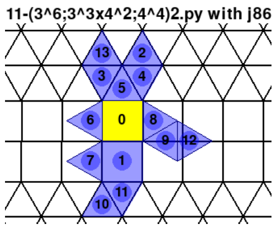
j86 $(3^6;3^3x4^2)1$ back			
j89 $(3^6;3^3x4^2)1$ back			
j90 $(3^6;3^3x4^2)1$ back			
j10 $(3^6;3^3x4^2)2$ back			
j50 $(3^6;3^3x4^2)2$ back			

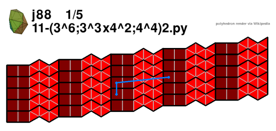
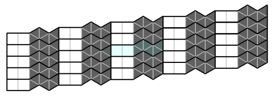
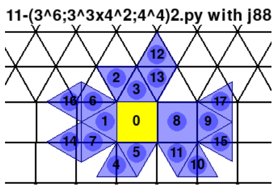
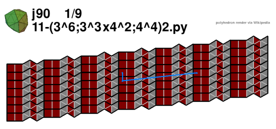
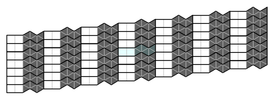
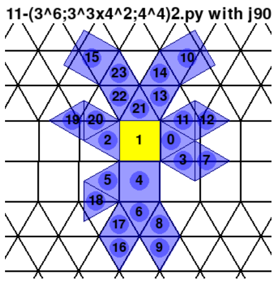
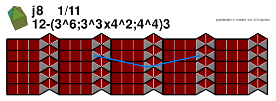
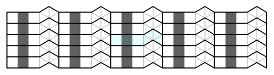
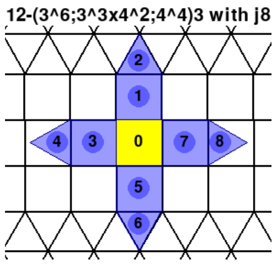
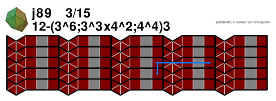
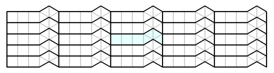
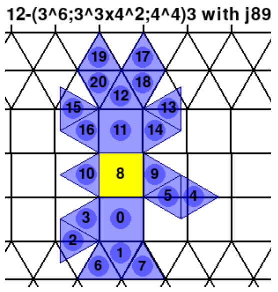
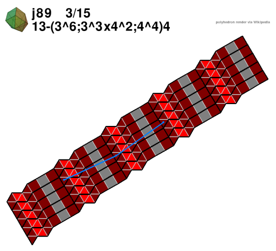
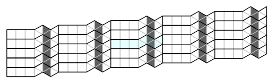
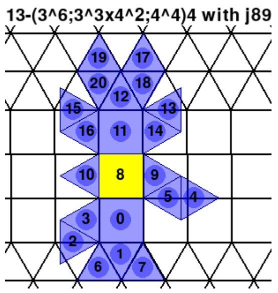
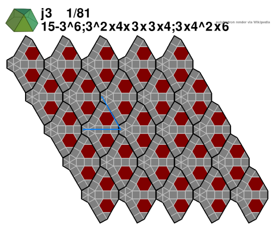
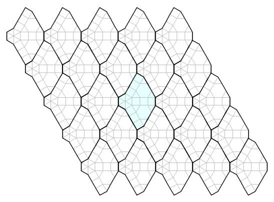
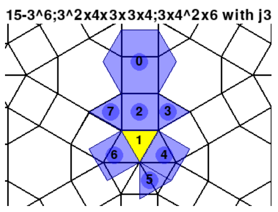
j86 $(3^6;3^3x4^2)2$ back			
j89 $(3^6;3^3x4^2)2$ back			
j1 $3^6;3^2x4x3x4$ back			
j10 $3^6;3^2x4x3x4$ back			
j50 $3^6;3^2x4x3x4$ back			
j85 $3^6;3^2x4x3x4$ back			

j86 $3^6; 3^2x4x3x4$ back			
j87 $3^6; 3^2x4x3x4$ back			
j89 $3^6; 3^2x4x3x4$ back			
j90 $3^6; 3^2x4x3x4$ back			
truncated tetrahedron $3^6; 3^2x6^2$ back			

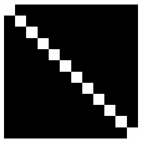
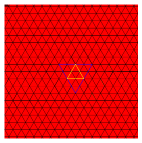
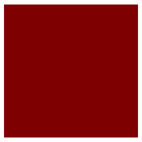
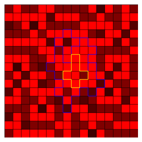
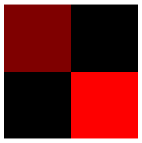
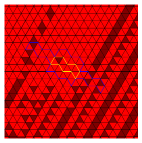
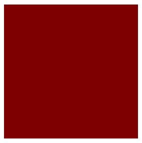
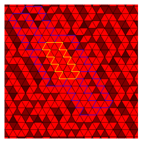
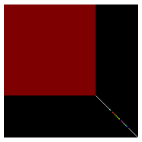
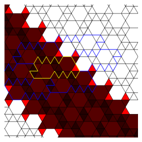
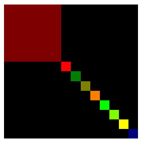
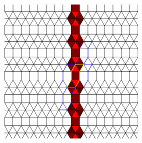
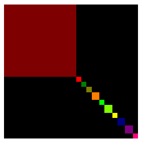
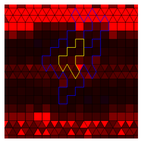
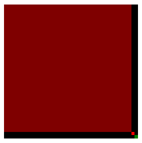
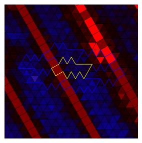
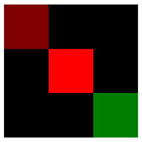
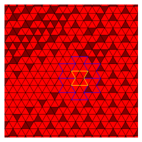
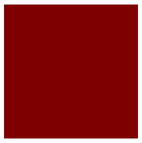
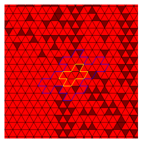
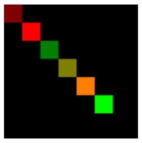
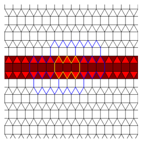
hexagonal antiprism $3^4x6; 3^2x6^2$ back			
j1 $(3^3x4^2; 3^2x4x3x4)1$ back			
j27 $(3^3x4^2; 3^2x4x3x4)1$ back			
j26 $(3^3x4^2; 3^2x4x3x4)2$ back			
hexagonal antiprism $02-(3^6; 3^4x6; 3^2x6^2)2$ back			
hexagonal antiprism $04-(3^6; 3^4x6; 3x6x3x6)1$ back			
j22 $05-(3^6; 3^4x6; 3x6x3x6)2$ back			

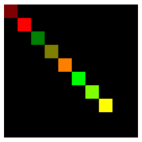
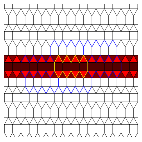
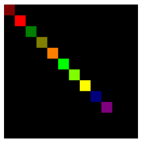
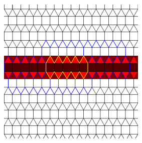
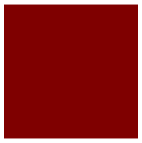
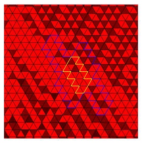
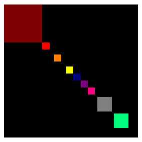
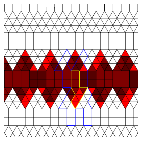
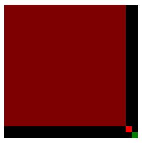
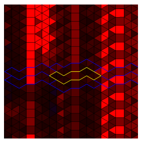
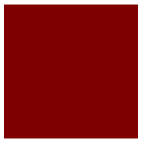
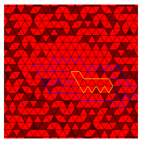
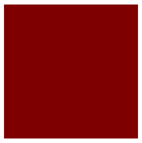
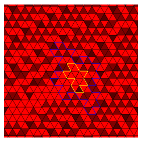
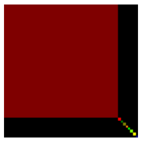
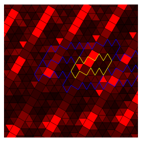
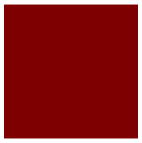
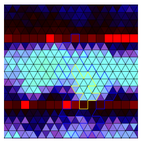
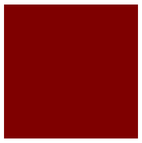
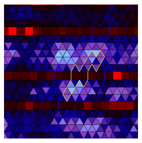
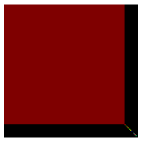
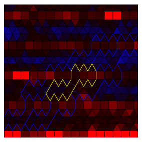
hexagonal antiprism $05 - (3^6; 3^4x6; 3x6x3x6)2$ back			
j22 $06 - (3^6; 3^4x6; 3x6x3x6)3$ back			
hexagonal antiprism $06 - (3^6; 3^4x6; 3x6x3x6)3$ back			
j1 $07 - 3^6; 3^3x4^2; 3^2x4x3x4$ back			
j10 $07 - 3^6; 3^3x4^2; 3^2x4x3x4$ back			
j27 $07 - 3^6; 3^3x4^2; 3^2x4x3x4$ back			
j86 $07 - 3^6; 3^3x4^2; 3^2x4x3x4$ back			
j88 $07 - 3^6; 3^3x4^2; 3^2x4x3x4$ back			

j89 $07-3^6;3^3x4^2;3^2x4x3x4$ back			
j90 $07-3^6;3^3x4^2;3^2x4x3x4$ back			
j8 $10-(3^6;3^3x4^2;4^4)1$ back			
j86 $10-(3^6;3^3x4^2;4^4)1$ back			
j88 $10-(3^6;3^3x4^2;4^4)1$ back			
j90 $10-(3^6;3^3x4^2;4^4)1$ back			
j86 $11-(3^6;3^3x4^2;4^4)2.py$ back			

j88 $11-(3^6;3^3x4^2;4^4)2.py$ back			
j90 $11-(3^6;3^3x4^2;4^4)2.py$ back			
j8 $12-(3^6;3^3x4^2;4^4)3$ back			
j89 $12-(3^6;3^3x4^2;4^4)3$ back			
j89 $13-(3^6;3^3x4^2;4^4)4$ back			
j3 $15-3^6;3^2x4x3x3x4;3x4^2x6$ back			

11.2 Tessellation Polyhedron rolling in their net tessellation

Tessellation Polyhedron	Roll Type	CFO adjacency	Preview
Tetrahedron	Plane		
Cube	Plane		
Octahedron	Plane		
Icosahedron	Plane		
Hexagonal antiprism	Band		
J1	Band		
J8	Plane		
J10	Plane		
J12	Plane		
J13	Plane		
J14	Band		

J15	Band		
J16	Band		
J17	Plane		
J49	Band		
J50	Plane		
J51	Plane		
J84	Plane		
J86	Plane		
J87	Plane		
J88	Plane		
J89	Plane		
J90	Plane