XiStrat Manual

Kai Brommann

XiStrat Manual
by Kai Brommann

Edition DRAFT 0.7.25
Published January 2010

http://xistrat.org

XiStrat is distributed under the GNU Lesser General Public License, Version 3; without any warranty, to the extent not prohibited by applicable state law. We distribute XiStrat **as is** without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. In no case unless required by applicable law will we, and/or any other party who may modify and redistribute XiStrat as permitted above, be liable to you for damages, including lost profits, lost monies or other special, incidental or consequential damages arising out of the use or inability to use XiStrat.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and we were aware of a trademark claim, the designations have been printed in caps or initial caps.

While every precaution has been taken in the preparation of this manual, the publisher assumes no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

Any statements in this document about future expectations, plans and prospects for the Group, including statements containing the words "believes", "anticipates", "plans", "expects", "will", and similar expressions, constitute forward-looking statements within the meaning of 'The Private Securities Litigation Reform Act of 1995'. Actual results may differ materially from those indicated by such forward-looking statements as a result of various important factors, including: the factors discussed in our most recent ToDo Report (a copy of which may be accessed through the Group's website), reliance upon strategic relationships, management of growth, the possibility of undetected software errors, the risks of economic downturns generally, and in XiStrat's industry specifically, the risks associated with competition and competitive pricing pressures and the viability of the Internet.

In addition, the forward-looking statements included in this document represent the Group's views as of the date of this release and these views could change. However, while the Group may elect to update these forward-looking statements at some point in the future, the XiStrat Group specifically disclaims any obligation to do so. These forward-looking statements should not be relied upon as representing the Group's views as of any date subsequent to the date of the release of this document.

Contents

List of Figures

2 Graph Construction

3 Playing Games

7 Case Studies

10 Knots and Links

11 Group Theory

15 Lattices

List of Tables

3 Playing Games

8 Graphs and Combinatorics

11 Group Theory

18 Chess Pieces Groups

B Third Party Software

List of Examples

List of Equations

Dedication

To our customers. They encourage us to go on.

Preface

This is a manual describing the idea and philosophy behind the XiStrat software package. Besides scientific background is provided :-).

Installation and usage instructions can be found in the Appendix A.

Part I

User Guide

First of all the XiStrat project is described in general.

Chapter 1

Introduction

1.1 Architecture and Idea behind it

In particular XiStrat (abbreviation of 'eXtended, enhanced and eXtreme STRATegy') is about turn-based, networked multiplayer strategy board games (without chance) on 3D-visualized polyhedra.

By the help of a computer-provided 3D representation (look at it as an MVC (model-view-controller) design pattern) it is quite comfortable for humans to play on those boards.

Persistency is provided by a relational database (offering networked access) when actually creating or manipulating the 3D data, besides the XiStrat server uses the database for storing all relevant gaming data.

XML is / will be used for communicating information about the graphs and the available matches to clients. Besides thereby (using files instead of a database) you are able to handle the graphs directly (work offline so to say) and import them into other tools.

1.2 Focus

Especially non-cooperative, zero-sum games with perfect information like Chess, Go, Reversi (Othello), n-in-a-row (Pente, Gomoku, Renju), Checkers (Draughts), Chinese Checkers (Halma), Abalone, Amazons, Arimaa, Dots-and-Boxes etc. will be dealt with. They are ideally suited for playing over the internet (low traffic).

Besides related areas (single agent, cellular automata, combinatorics, graph/group/complexity/-knot theory, quasicrystals, discrete geometry, computational algebra, mathematical physics) are discussed as we occasionally encounter them, so actually the XiStrat project belongs to a wider field of general recreational mathematics.

1.3 Feedback

Of course help, comments, hints, contributions and critics are welcome. We wouldn't like to reinvent the wheel, please tell if you are aware of prior art. Any feedback is appreciated. As you have seen there is still some space left on our HomePage for new success stories. Well, please don't take them too seriously though, they are mostly just enthusiastic, passionate or polemic FUD and provocative advocacy. In case the project may appear to be stagnant, don't contemplate but give a helping hand!

And anyway please excuse the in some places of this manual rather informal tone, it is not ment to be offensive but motivating.

1.4 Thanx

Thanx to all upon whose work we have built the XiStrat project, which of course will for ever be a 'work in progress'.

1.5 Epilog

To be honest we intend a scientific approach to the strategy games. The generalisation to non-standard boards (curved 2-dim surfaces) motivates some flexibility in the brain (what can't be harmful). And a right theory must apply to such general cases, so in working within a broad framework we avoid too short-sighted ideas directly, those won't be the correct solutions in the end anyway. And just in case we intend to model dynamics by nonconstant geometry, sooner or later we need a general setting anyway.

Last but not least it gives appealing pictures for a non-expert audience as well as many welcomed starting points with already established connections to modern mathematics.

In the long run the project could lead to new insights in music, football (soccer, rugby), economics, Wall Street as well as politics. Hopefully it can help to find solutions other than Nash-equilibria. The latter are worst case scenarios, a fact that seems to be not widely known.

And in the future (as all science) such a theory of war games might be applied in military reality, of course let's keep that in mind as well right from the start and avoid uncomfortable surprises later on.

1.6 Formal Stuff

See the TODO file about known bugs and issues.

1.7 ToDo

Things still to be done are mentioned at the end of each chapter. This is starting right here:

- set up a XiStrat demo server running 24/7
- even more elaborate use of MathML
- customize the stylesheets for a documentation having some XiStrat logo/look-and-feel

Chapter 2

Graph Construction

2.1 Introduction

We deal with special polyhedra: (orientable) graphs (embedded on surfaces) also called maps. Boundaries are allowed. Every face can be reached from another in a couple of steps.

The polygons have a constant number poly (3, 4, 5, 6, . . .) of vertices/edges (s.th. like face-valence). Actually things do carry over to general polytopes with variable valences (the least common multiple will then replace poly).

The icosahedron for example is furthermore 'face-transitive' in the sense that all the faces have the same status. But this is not the case for all graphs treated here though, the dual's valence may vary.

As already mentioned a relational database MySQL as well as XML are used.

The Java3D scenegraph concept enables us to concentrate on our specific application instead of fiddling around with underlying layers such as OpenGL.

The 3D - GUI provides the following feature: you can use the mouse buttons and rotate (left mouse button), translate (right button) and zoom (middle button).

2.2 Creating 3D data

You can create new graphs within XiStrat. A spring embedder (see [314]) with repulsive forces between the vertices and attractive forces along the edges is used (the constant parameters for these forces might be subject of further tuning). Probably it would be a good idea to add another repulsive force, this time between faces, thereby punishing a layout where two faces co-planar onto each other, this would help for example in case of some variants of the morphed icosahedron when there is only one common shared edge. Another useful feature would involve curved edges (non-straight) and faces (non-flat).

Please refer to Appendix A for some installation and usage description.

In general it is possible that faces (except the triangle case of course) are not flat (i.e. in one plane). The triangulation (needed because natively only trias and quads are supported having drawing primitive type support in OpenGL implementation) is done using poly elements and takes thereby care that later on pieces can properly be drawn by using special points (center, . . .) of the polygons.

Sometimes (actually quite often if the graph gets complex) the spring embedder will return data which represent suboptimal solutions. Then simply try again. One cannot ignore the fact that in general this sort of graph layout falls into NP-complete, see Chapter 19. The algorithm is non-deterministic. It is random Monte Carlo (may return an incorrect answer with probability of error bounded by user) and not Las Vegas (i.e., guaranteed correct output) so far. Besides there are cases where multiple results should be regarded as allowed solutions, that is rendering without unnecessary self-cutting (and little stress or energy in the overall situation).

The general procedure needs two files:

2.2.1 Infiles

The infiles' syntax is similar to the specication in VRMLGraph. You simply tell pairs of connected vertices (please start with the vertex id 0). But since we have to deal with some corner cases as well, we allow loops (edges connecting a node to itself) and multiple edges with the same ordered pair of nodes (moreover at the same time possibly connecting thereby the same set of faces). Only the full information consisting of both ordered pairs (of points as well as faces) specifies the edge label later on.

A modest warning now: Internally we look after the highest used vertex (node) label (so the labels should be integers) to estimate how much memory for vertices, colors and so on to allocate. In case you get strange rendering results with faces half-dead or dark, then perhaps you have used too high labels and left too much space unused in between. For big scenes this can happen and be useful to keep the overview, so perhaps our Create 3D Data Tool should be rewritten to avoid using the labels but instead the actual index of the vertices. At the moment just use the resources with care. A sort of soft limit is reached at about 5000 vertices so far.

2.2.2 Property Files

The CreateData tool must be told the expected number of vertices respective edges a regular polygon has got. This can't easily be figured out automatically (girth and face-valence are not always the same). And because then we have property files anyway, we tell about the number of overall polygons (invalid included) as well. In principle the polygons might be found out by machine itself though.

A possible improvement could be to directly specify in the input file what vertices form a face, because at that time one probably already has those faces in mind, and this way the faces don't have to be reconstructed again later on by looking at nothing more than the vertex adjacencies.

The "Rubik" cube variant with intermediate states need a lot of topologically different graphs, and the needed infiles are autogenerated.

2.2.3 Faces and Orientations

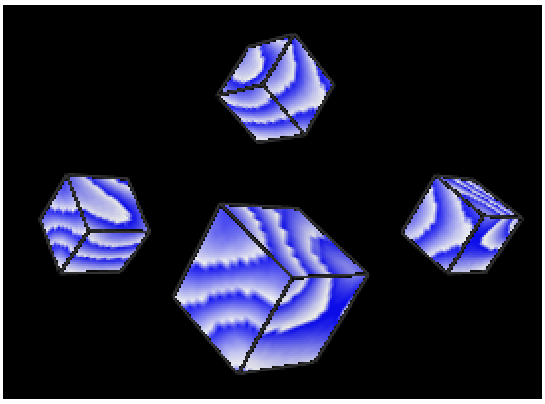
After creating 3D data, an algorithm is used to find out about what vertices are connected by a circle of given length, so they form a face. Besides for each face the vertices' order should specify an orientation with the resulting normal pointing outwards.

2.3 Colorings

In the next step a coloring of the faces is done (see Section 8.2 for more information).

2.4 Procedural 3D Textures

One can apply texture to objects by literally carving them from a 3-dimensional (marble, wood, stone) object to avoid the suffering from 2D warping mismatches (see [313]). Here is a screenshot:



using Perlin noise

Figure 2.1: Perlin noise example

2.5 Autogenerated Files

The needed files can be automatically created for 'Rubik' like moves resulting in new board variants.

Besides our construction of quasicrystals uses autogenerated files. Dependent on the graph name the substitution rules for Penrose or Ammann-Beenker [128] are applied on the rhombic tiles.

2.6 ToDo

Here is a loose collection of proposals:

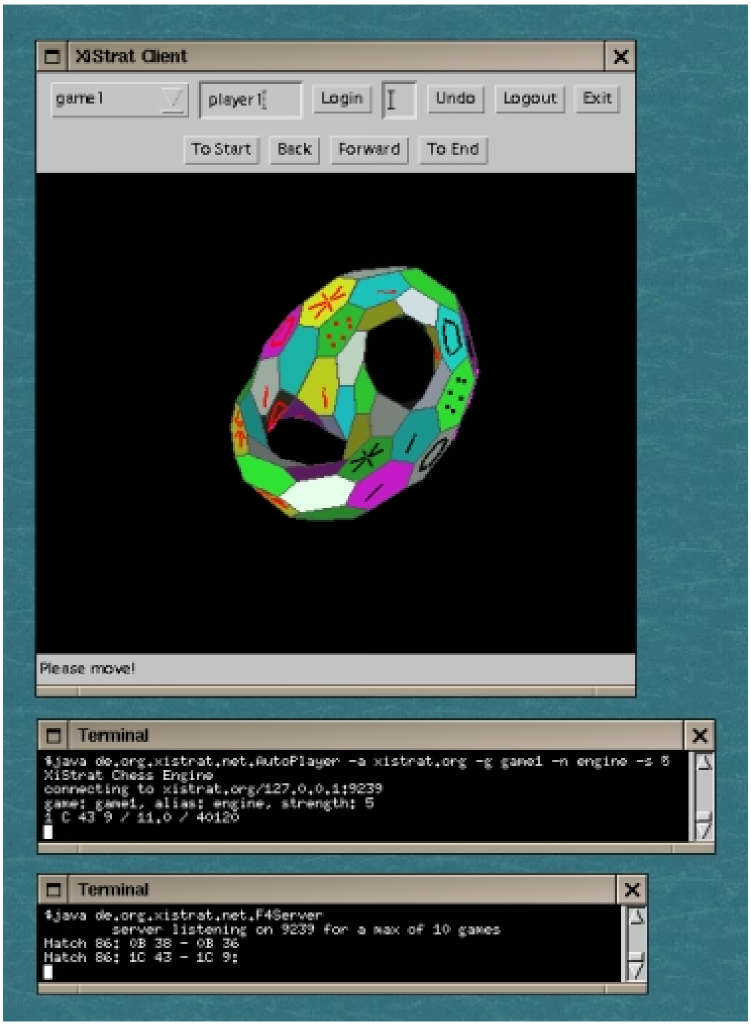
- interactive (editor for) graph creating: drag / drop and copy / paste
- at the moment the mirrors have fixed angle, size and position. they should be mobile, allowing the user to specify just about every desired view - point (at the moment you can move the mirrors around and zoom them, but this doeesn't change the transfo by which they get their data so far)
- GraphML or VRML interface (.wrl) for im-/export
- having sql face tables with n=3, 4, 5, 6, 7 ... columns when creating 3D data is really ugly; there should be a possibility of better normalform design
- more surfaces: for example see references (discrete minimal surfaces, Laves nets, Poeppe polyhedra, stellations of higher order, more 3 dimensional zonohedra and zonotopes, hypercubes, projections from higher dimensions), and s.th. like an '11-hole torus' with a 5-fold rotational symmetry resulting in a covering space M_{11} -> M_3 (see [30])
- quasicrystals higher iterations
- the boards might get a nicer 3D texture to look really like made-of-wood, and the stones must have texture as well
- various mathematical aspects of the graph layouting
- using more advanced Java3D features (shared vertices data etc.)
- autogenerate dual to given graph (result may be a polytope)
- for all graphs use some inflation to get a triangulation like the oct_stl_II example (Coxeter reflection group with fundamental region and so on), and then do deflation of the 4 triangles around any edge to get another poly=4 graph (line graph with edges as faces) (GRAPE can do edge graphs as well)
- again for all graphs use the paper model atlas thing (the cons/cuts from what once has been -q -s 4) to construct corresponding flat graphs with cuts as boundary edges
- for cartographic L2_7 (here moreover a triangle has got itself as a neighbour, because two edges are connected to each other), M12, J2 etc. we would need edges as loops (a simple circle starting and ending at the same vertex, which should be approximated by many small triangles from the loop the another vertex (the layout procedure would need a tuning as well, but in principle there is no reason not to do it)
- instead of working directly on the JDBC API to achieve persistence, one cound make use of for example JPA or take another standardized approach

Chapter 3

Playing Games

3.1 Introduction to our Internet Games

This chapter is about the sportman's point of view.



Please move!

```
$java de.org.xistrat.net.AutoPlayer -a xistrat.org -g game1 -n engine -s 5
XiStrat Chess Engine
connecting to xistrat.org/127.0.0.1:9239
game: game1, alias: engine, strength: 5
1 C 43 9 / 11.0 / 40120
```

```
$java de.org.xistrat.net.F4Server
        server listening on 9239 for a max of 10 games
Match 86: 0B 38 - 0B 36
Match 86: 1C 43 - 1C 9:
```

XiStrat server, engine and client GUI in action

Figure 3.1: *The Overall Architecture*

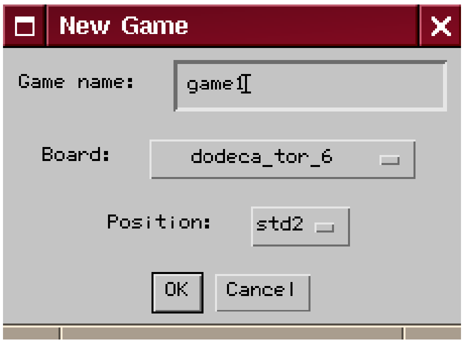
The client - server framework (see Figure 3.1) uses connections through TCP/IP sockets. For the moment XiStrat is based upon (and therefore behaves in many ways like) the forza4 software (see [304]). The server spawns separate daemons for each game and uses a thread per player approach. In the future perhaps a more generic internet client - server communication framework will be implemented, but for now forza4 completely serves our purposes.

So far only the Chess and Go variants are existing, the following documentation deals only with these particular families. But of course the other variants will deliver new insight into old games as well as soon as the implementation is done.

Now first of all the following quite useful feature should be mentioned: by **CTRL** - pressed left mouse clicks in the (client-side) GUI you can find out about a face's id. And don't be confused by the PASS button in the client panel, it is actually only functional in case of the Go game.

3.2 General Procedure

At the moment up to 4 different players are supported (black, red, blue, and green color), but this number is easily increased. How many players are participating in a particular game depends on the selected starting position.



Game name: game1

Board: dodeca_tor_6

Position: std2

a new game is started (old layout)

Figure 3.2: *New Game*

After choosing the game family (variant), (see Figure 3.2) a new game is created by entering its name and which board a game is to be played on. One can select from a correspondent list of positions to start with. A starting position contains information about the board, the positions of participating pieces (may be none) and what color actually is to start.

Once a new game has been created, the players can join (login). The first player logging in to a game will get the party with black colors, the next will get red, and the third blue. The the game begins, turns go round in the same order.

3.3 Chess Variant

The standard positions (std0*) are not supposed to make sense from a sportive gaming point of view, but simply provide all types of pieces for testing purposes. Of course the really interesting positions are yet to be found - contributions are welcome!

For every party, the piece having the king role is indicated by a magenta 'circle' shape around it. That specific color was used for royal clothes in the middle-age and so is most suitable for our purpose.

The pieces can move as described in Table 3.1 (in the GUI keep the **SHIFT** key pressed, select the piece and then the desired new location with a left mouse click).

At most one piece may be located at a given space/time. Piece D is not transmutating (promotion) nor being reflected but just can't move anymore when coming to a playfield boundary.

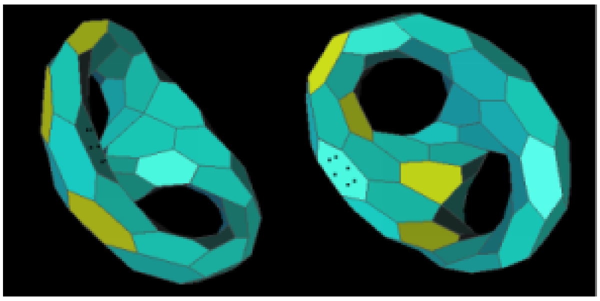
At the moment the implementation of piece E is missing in the GUI (it would require two clicks).

On one hand there are well-definable moving rules, on the other there exist unique ways to make up visual representations for the pieces. The mapping ψ between those two systems actually transfers a structure from an algebraic system (see Section 11.2 if you need s.th. to go from algebra to the piece movement) to a set, and then by means of ψ one can define a structure on the set, and this finally makes ψ a morphism.

Because visuals tell more than thousand words, Figure 3.3 demonstrates how a piece of type C can move.

type	rule
A	1 step to a neighboured face
B	like A, but infinitely many subsequent steps, having +-1 edge between last and next visited face
C	like B, but only 1 move with distance of 2 steps (jumping)
D	like A, but only one direction, direction evolving with +1 edge (counter-clockwise) in between
E	like D, but no restriction upon the next direction
F	like B, having +2 edges in between (asymmetric)
...	...

Table 3.1: *Piece Moving Rules*

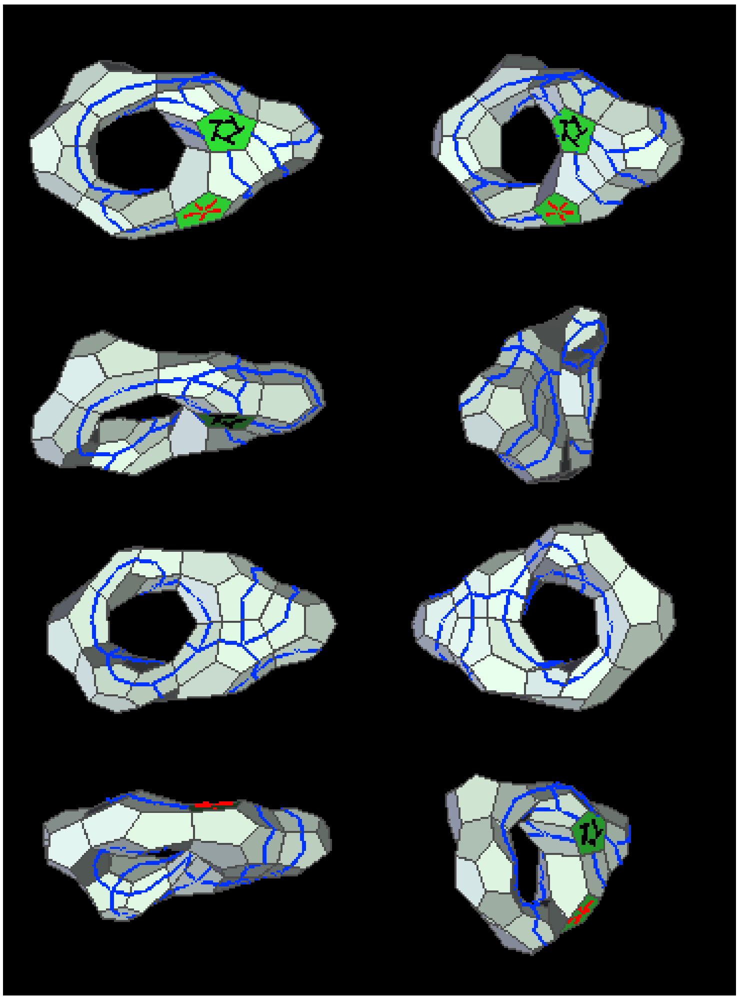


About how a piece of type C can move on a board with hexagons (best seen in colour).

Figure 3.3: *Chess Piece Type C*

In the following picture (see Figure 3.4) you see how a black piece F can move while another piece, a red A, coexists. The two pieces have been placed onto green faces for didactic purposes only. The blue lines represent directed ways the piece F can travel along.

Keep in mind that it's an asymmetric case this time (and so the visual representation does of course indicate that), but this only becomes measured as soon as another piece enters the scene and starts disturbing, that is to say the blue line showing how the red piece can be captured is not in effect behind the face occupied by the red piece, and so the face between the two pieces wouldn't be reachable by the black F within a single move if it wasn't along another line.



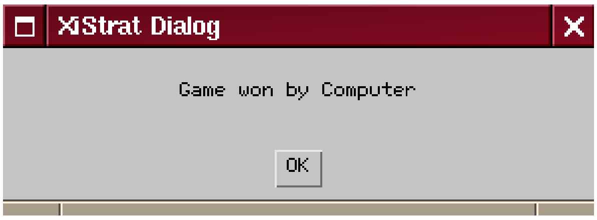
A piece of type F on a board of pentagons whilst showing the effect of a second piece.

Figure 3.4: *Chess Piece Type F*

One word of caution: One should ensure (for example) that in case where a moving rule definition collapses (or becomes unclear) in a special case (e.g. poly=3 or others), the correspondent (mapped) visualization rule should collapse (or . . .) as well. This field is still under construction and therefore things will certainly change over time. For example, the piece F (with distIE = +3 asymmetric) moving rule becomes symmetric in case of poly=6 (for 3 = 6 - 3), but the current visualization stays asymmetric, so it's not perfect at the moment.

Btw. those movement lines look like geodesics (or autoparallels) and a torus knot or link thing! See the Chapter 10 chapter for more information. Or, actually it strongly resembles the curl-free (Pólya) vector potential field (rot grad = 0), and we have no problem with considering a related picture for the div rot = 0 (just draw lines along faces of same distance from the piece origin, this has to do with theta series as well).

Rules are adopted from Chess. Pieces with a royal role should be taken special care of (see Figure 3.5).



XiStrat Dialog

Game won by Computer

Your king has been at stake.

Figure 3.5: *After a Checkmate*

We do not stop in case of a traditional mate position, but anyway some legal move must be done by the losing party, and then we require the king to be captured to finish things off. To simplify matters, at the moment stalemate and checkmate both are counted as a win.

3.4 Go Variant

We have developed a XiStrat Go variant. Because the task of rendering stones placed on a boundary probably wouldn't be well-defined, and anyway since we appreciate a uniform codebase for our game framework, we decided to have the stones sitting inside the faces (and not on the vertices). This may need some time of accomodation for experienced Go players, but from a mathematical point of view it's just using the dual graph and no principal difference.

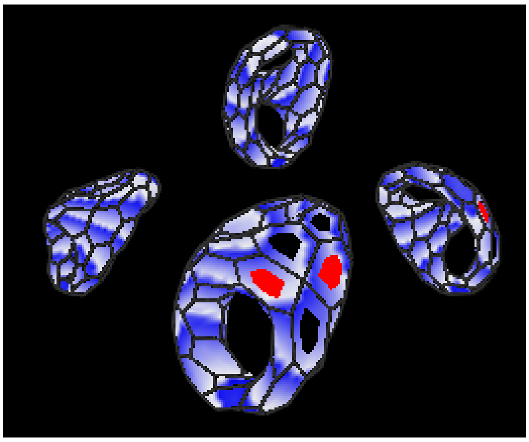
In the game Go, one can see much more 'sente' when played on a triangle board, whereas for example on hexagons there are plenty of initial liberties.

We prefer the Japanese rules (at the moment, leaving some subtle bestiaries out for now, that is to say counting territory, prisoners and taking Komi into account). It is aesthetically appealing, because (under area scoring such as for example Chinese rules) unnecessarily placing a stone inside one's own safe territory is not punished, whereas it's an indication that maybe the player is not sure about the state of affairs. Supporting such slack play throws away a possibility to distinguish the level of play. But perhaps s.th. like the Tromp-Taylor set of rules would be easier to deal with from a mathematical point of view.

The stones can be put in the standard way (in the GUI keep the **SHIFT** key pressed and select the desired location with a left mouse click).

After all parties pass in a row, the game is over. Then dead stones must be removed by all players (again using the **SHIFT** key and a left mouse click on every dead stone), then all players pass again, and finally somebody has won (the removing phase of dead groups could be avoided once we can have the declaration been done by the computer).

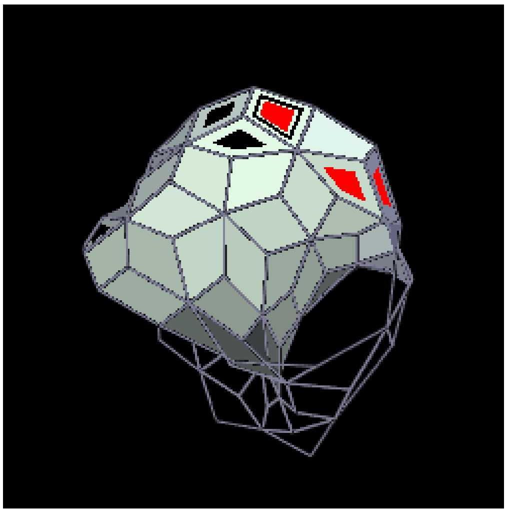
In Figure 3.6 we can see a 2-players game on a marble 'Hexa Tor' graph.



Go on a marble board (in mirror view)

Figure 3.6: *Go on Hexa Tor*

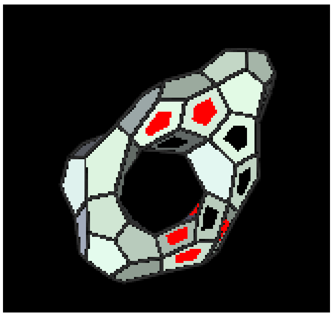
Below you can watch an ongoing simple Ko fight (Figure 3.7). An immediate recapture is not allowed as indicated by the black square surrounding the red stone.



Ko fight on a quad-hex graph

Figure 3.7: *Ko on a boundary*

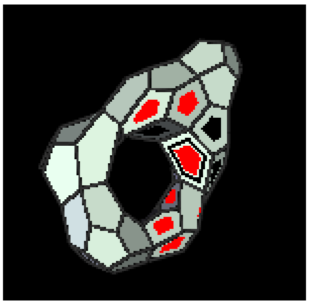
Here we watch in sequence a 2-stage Ko on the "dode_tor_6" graph (first of all the initial position, the red party to move).



2-stage Ko fight on a dode_tor_6 graph (initial situation)

Figure 3.8: *2-stage Ko 0*

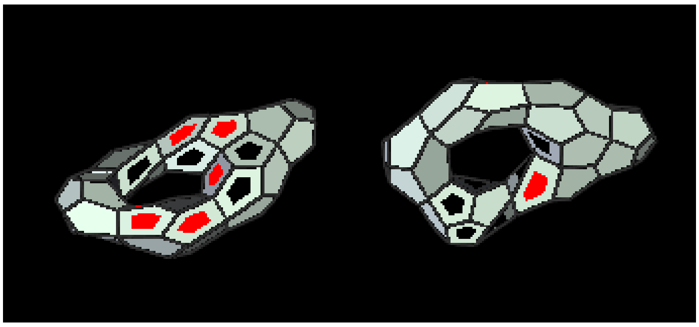
After the first capture by the red stone (Figure 3.9).



2-stage Ko fight on a dode_tor_6 graph (after the first capture)

Figure 3.9: *2-stage Ko I*

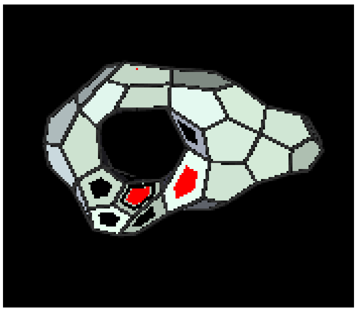
Then black plays a Ko threat somewhere else on the board. Red could answer this, then black would perhaps recapture. But the red party decides to ignore it, and the situation looks like Figure 3.10.



2-stage Ko fight on a dode_tor_6 graph (after first Ko threat)

Figure 3.10: *2-stage Ko II*

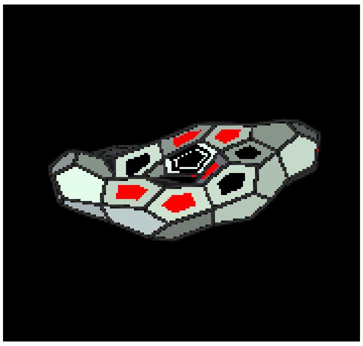
Red has taken the second Ko (Figure 3.11).



2-stage Ko fight on a dode_tor_6 graph (after taking the second Ko by red)

Figure 3.11: *2-stage Ko III*

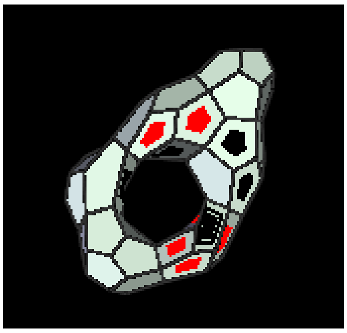
Black now plays another Ko threat somewhere on the board, and this time red answers there (instead he could connect). Black can now recapture the Ko (see Figure 3.12).



2-stage Ko fight on a dode_tor_6 graph (after retaking the second Ko by black)

Figure 3.12: *2-stage Ko IV*

Black ignores a red Ko threat and takes the initial Ko (see Figure 3.13).



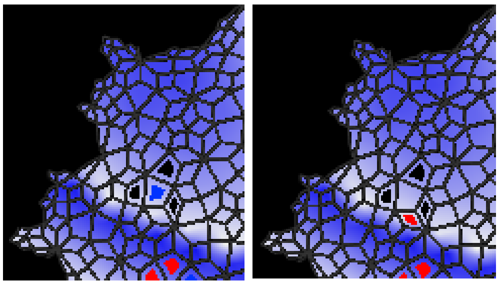
2-stage Ko fight on a dode_tor_6 graph (after second retaking by black)

Figure 3.13: *2-stage Ko V*

The fight may rage back and forth when now red plays a threat, black answers it (instead of connecting), and red takes the first Ko again.

Generally speaking bear in mind that on this graph a Ko possibly cannot arise in some places where the faces are connected too tightly.

Here we see a match with three parties involved. The black and red players seem to cooperate, at least they have decided to remove (in alliance) a blue stone from the board Figure 3.14:

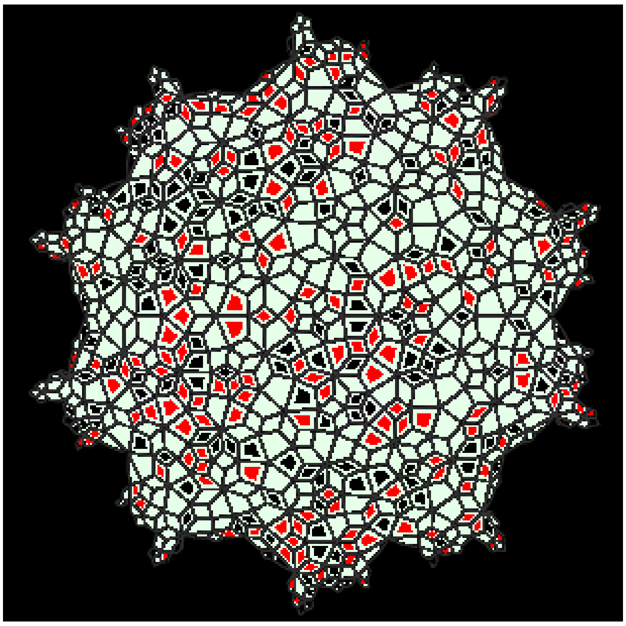


Go as a multi-player game (three parties)

Figure 3.14: *Go 3 players*

It's counted as a loss for the blue party (and not as a gain for red what would be another option of course).

Finally we finish our sight seeing tour and watch a wild fight between two random engines on a quasicrystal (Figure 3.15).

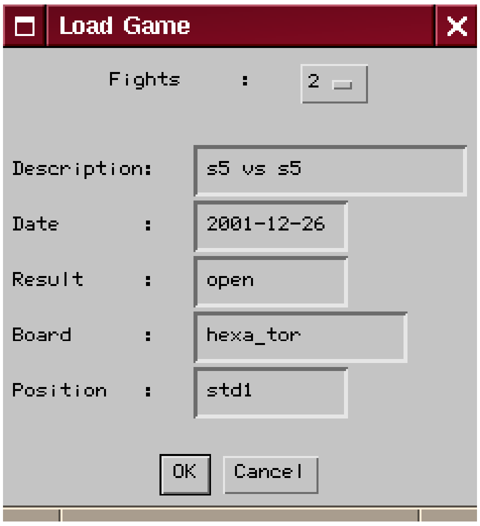


random Go on a quasicrystal graph

Figure 3.15: Random Go

3.5 Still more features

You can load games (Figure 3.16). Ssome demo games are included in the distribution, and new games are automatically recorded on the server side.



Load Game

Fights : 2

Description: s5 vs s5

Date : 2001-12-26

Result : open

Board : hexa_tor

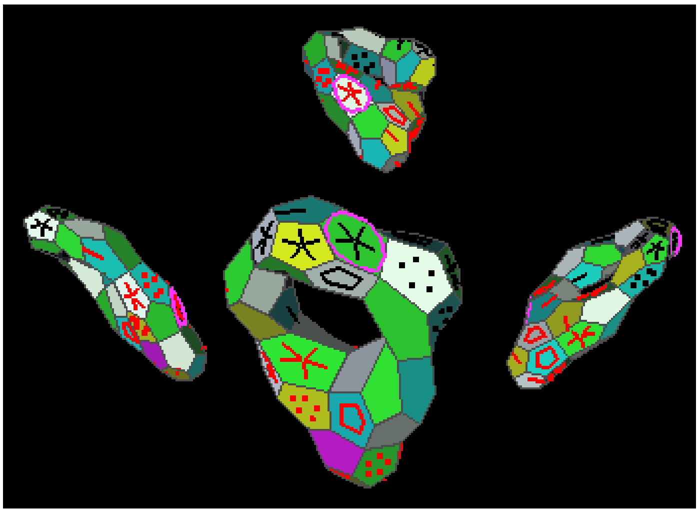
Position : std1

Loading a game for replaying.

Figure 3.16: Load Game

For replaying simply use the 4 buttons on the history bar. Besides you can have the current position saved into the database.

As described in Appendix A, you can have additional helping mirror views in the Canvas3D. This should make it easier to get an overview about the scene. Of course still depending on the actual concrete board, it is now possible to have all faces in sight at the same time, so you don't have to rotate the scene all the time during play but just try to identify the corresponding original and mirrored parts of the board. Here is a screenshot (Figure 3.17):

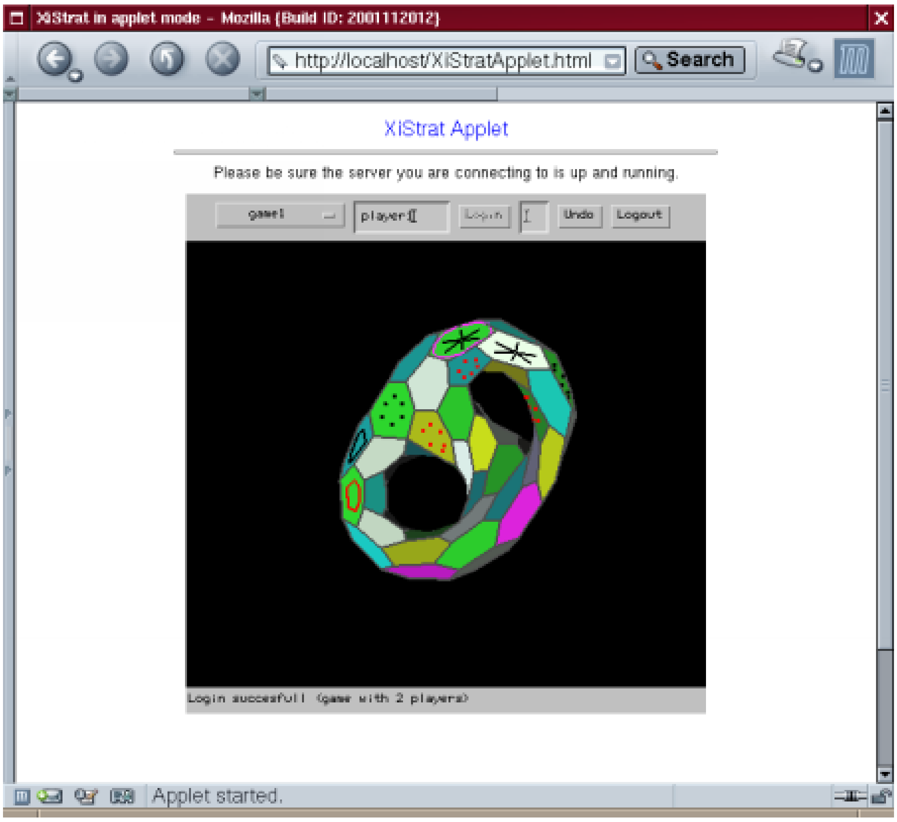


XiStrat client with mirror views.

Figure 3.17: *Client Mirror View*

The object at the middle/bottom is the original playfield, the three others of slightly smaller size represent mirrors showing the backside under a varying point of view.

In the screenshot Figure 3.18 we see how the future might be.


XiStrat in applet mode - Mozilla (Build ID: 2001112012)

http://localhost/XiStratApplet.html

XiStrat Applet

Please be sure the server you are connecting to is up and running.

game1

player1

Login successful! (game with 2 players)

Applet started.

using the browser plugin

Figure 3.18: *XiStrat Client in Applet Mode*

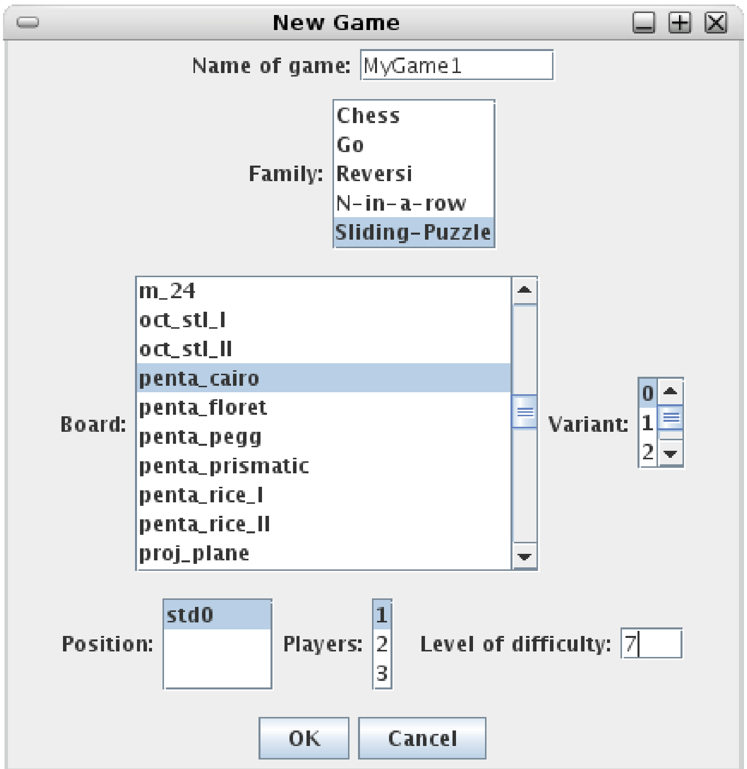
Another platform might be mobile devices (with VGA display, see for example the OpenMoko Neo1973).

3.6 Single Agent Games

These are realised in a networked architecture as well for storing time records (hall of fame) purposes and so on (and perhaps one can come up with some related multi-player game versions later on).

3.6.1 Sliding-Puzzle

We implemented a generalization of the Loyd’s 15-16 puzzle (with full support of features like history, undo and helping mirror view of course, but no storing or loading of games).



New Game

Name of game: MyGame1

Family: Chess, Go, Reversi, N-in-a-row, Sliding-Puzzle

Board: m_24, oct_stl_I, oct_stl_II, penta_cairo, penta_floret, penta_pegg, penta_prismatic, penta_rice_I, penta_rice_II, proj_plane

Variant: 0, 1, 2

Position: std0

Players: 1, 2, 3

Level of difficulty: 7

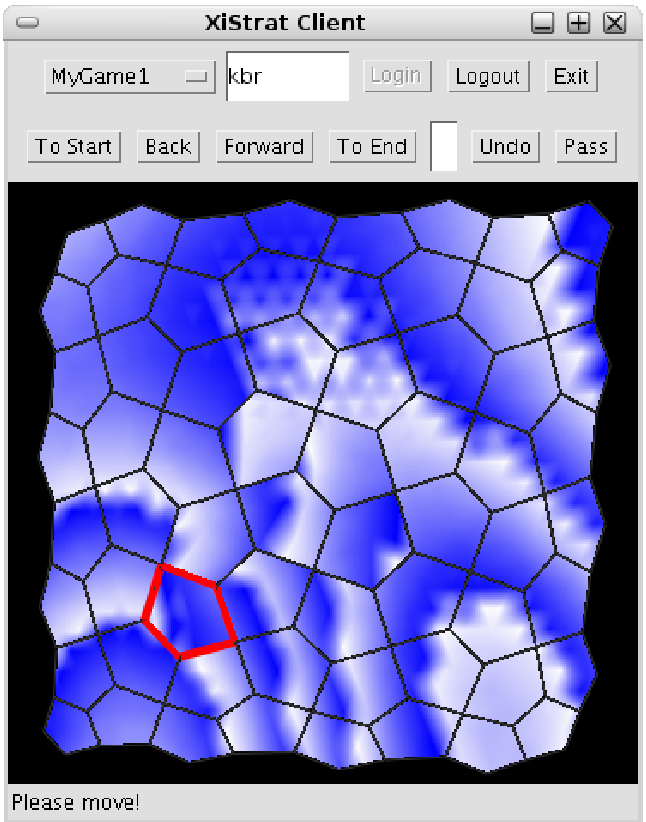
starting a new sliding game

Figure 3.19: New Sliding Game

The number of players is bound to one, what makes sense for single agent games.

Whereas the screenshot above indicates that 7 subsequent random permutations are wanted, in what follows we decided to lower the level of difficulty a little bit (to 5) because this here is meant to be an introduction for beginners.

The face framed in red colour is where the slider is located. In the standard physical incarnation this would be empty (because some other tile must arrive there), but realized as a computer game we are free to make use of this possibility to visualize as much information as possible.



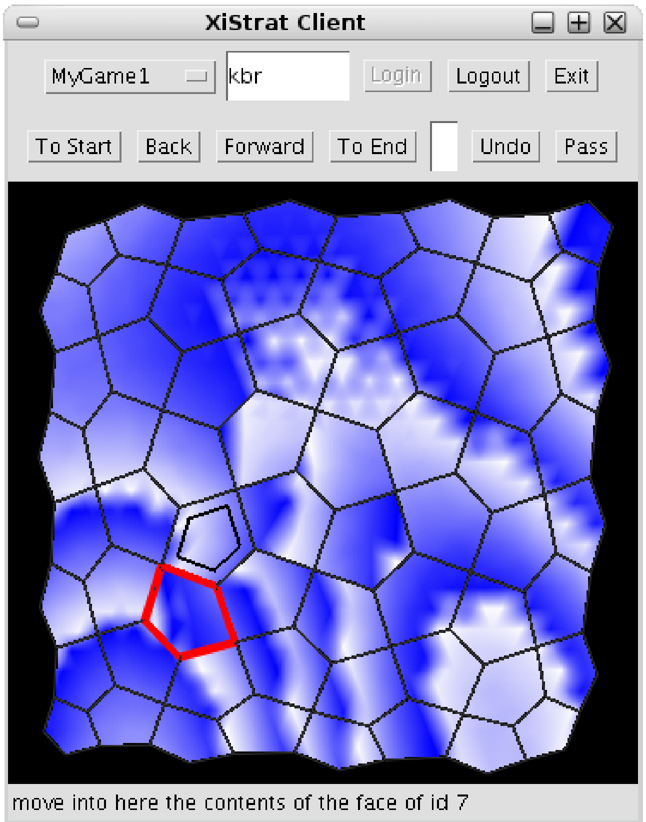
five slides away from the success

Figure 3.20: *Sliding Puzzle Initial Position*

You can depict a slight misplacement of some textured stones near the red thing, don't you?! Instead of numbering the faces somehow we simply use the texture to indicate the desired outcome.

Now by left-mouse clicking on faces (which must be neighboured to the slider of course) the slider moves and the contents of the two faces involved is exchanged. Hereby a anticipation of a straight direction is involved and hard-coded as poly/2 (you will feel it in case of odd poly). Another possibility would be to make use of our boundary-lines creeping around corners (providing a direct link to our groups btw.) but we decided to keep at simple (that is like in the original 15-16 puzzle on a square 4x4 board).

In order to help the user a little bit since the texture might look similar every here and there), by lect-mouse clicking on a face while the AltGr key is pressed (we use AltGr simply because SHIFT for actual slide and CTRL for id-telling are already taken, and Alt just didn't work:-() a mark is placed on the face the contents of which must be slided somehow into this clicked place (the texture will tell about the orientation). Here is an example where a click on the red-framed face was done:

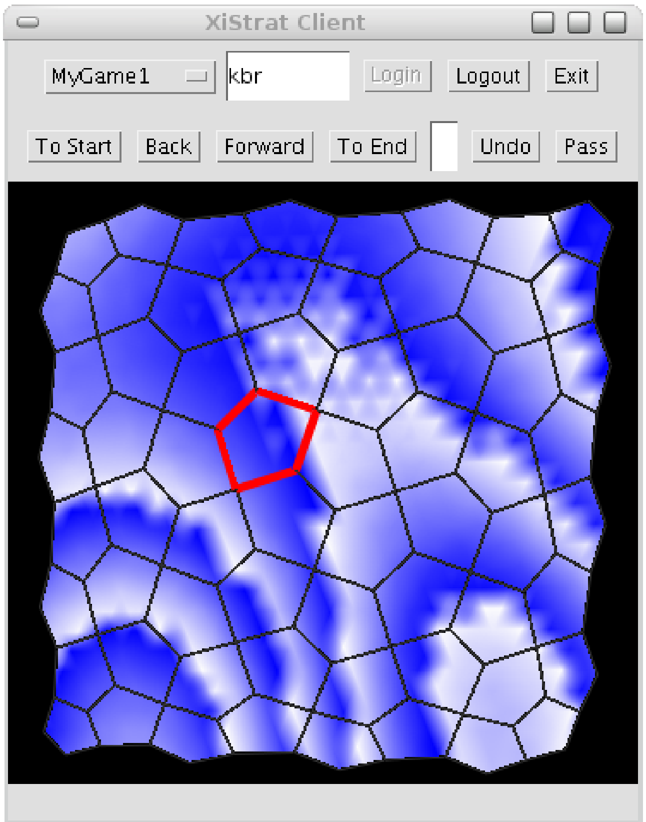


move into here the contents of the face of id 7

left-mouse click together with AltGr

Figure 3.21: *Sliding Puzzle Help*

Finally we have managed:



the texture indicates that the solution has been found

Figure 3.22: *Sliding Puzzle Solution*

The congratulations are well deserved.

A mathematical treatment could try to estimate the optimal number of moves needed for a given permutation, or provide invariants whereby unsolvable starting positions can be detected, or the positions with the longest distance to the solution.

Further generalizations could be sliders with more complicated moving rules (instead of Chess piece of type A let's say some jumper like type C), or perhaps a two-player version, with only one slider (but no simple reversion of the opponents action), or perhaps even better with two sliders and goals still to be defined.

YOU HAVE WON!

the sliding puzzle is solved

Figure 3.23: *Sliding Game Congrats*

3.6.2 Rubik Game

Still to be done. Here screenshot showing GUI in action, how to specify area to be rotated etc. will appear in the future.

One has got to restrict oneself to subgroups of the full group making the problem interesting to solve. And you are invited to study the graphs related by morphing (see Section 7.24). See Section 11.4 for a group theoretical treatment and Chapter 6 for a sophisticated auto solver (chains of subgroups leading to recipes), but it's mostly vaporware at the moment.

3.7 ToDo

Here again is collection of proposals:

- HTTP-tunneling
- clocks and time constraints
- observe other games feature, chatting and sound effects
- for general networked multi-player board games a generic framework is needed (perhaps being some GTP superset in its Go instance, so already existing tools for regression testing and whatsoever could be used)
- allowing the clients to address the mysql server directly is perhaps a little bit too sophisticated :-)
- XML for games notation (investigate Chess PGN and EPD, Go SGF, GGF)
- more sorts of pieces: imagine new symmetric nice intuitiv graphical display ways as well as good ways to define moving rules; break symmetry (looking from inside / outside); pieces occupying more than one face at a time
- perhaps make pawn (piece D) capture diagonally (forward plus to the left and right) (i.e. not in the same way as it moves), then s.th. like closed positions in Chess could arise (where knights become valuable); up to now the piece D only get's stuck when coming to a boundary or friendly forces, and not when arriving at an enemy ... this is losing some spirit of the game
- regarding the idea of a bishop-like type of piece: instead of keeping the amount to be rotated constant ('a' is flip and then a rotation of poly/2), let's vary the rotation (to poly/2 +-1 for poly even, poly/2 and poly - poly/2 for poly odd) alternatingly, whereas probably both possibilities to start with should be allowed, this would provide another long-mover type of piece, let's call it G, it is still on average walking more or less in a straight manner; and then we add a jumping element (step size set to 2 like in case of the type C), and finally this could by the new type H and coincide with the definition of a bishop on a traditional board
- pieces like B and F might be allowed to be reflected at a boundary and then (in contrast to traditional Chess moving rules) continue their travel within a single move
- make it visible to other players what has been last move of opponent (highlighting or s.th. like that)
- undo feature might be improved by having it incorporated in the history browsing (you go back, and then simply start playing from some old position again, and this could be interpreted as undo)
- Go three parties territory scoring is probably broken
- draw, stalemate, bestiaries
- generalization from 2 to 3 dimensions (3D Chess)
- implement the Rubicon game on icosahedron (Rubik game analog variant). Example standard 3*3, though not only the 90° turn, but 1/3 of it. prerender VRMLgraph such new graph, then Java3D alpha morphing (this will look nice animation of the move) (thereby this ugly intermediate state is avoided where one can see the inner things of the cube). It's actually a moving along closed edge lines, and thereby the faces colors contents are moved along then (quite plastique). One can put torsion stress energy into the connections (genus > 0), so has to do with this proposed editor for graph construction as well. looks great, am sure! See what vertices and edges should be displayed fixed, and what parts will have to change place due to a move
- some puzzles like Hi-Q (also called Solitaire) etc. on these discrete manifolds
- one could display some lines (orbits), and the game idea then is to do discrete Reidemeister moves to unknot them or (when this is impossible because there is some inherent knotting) at least reach a comfortable state where the remaining crossings are really unavoidable

- real-time games (instead of turn-based) on those boards
- another Chess variant could allow the re-inserting of captured pieces (as in Shogi)
- or one could get rid of the king role in the Chess game to avoid as much rules as possible (just like in Checkers: if at least one piece survives of each party, then it's a draw, and only if all pieces are captured, the game is decided)
- yet another Chess variant could allow multiple kings of the same party, and as long as at least one of them is still alive the party is considered alive as well
- Atari-Go (first capture wins)

Chapter 4

Problems, Compositions and Studies

4.1 Chess problems

Problems (compositions, studies) represent the scientific and the artist's view on the topic. All composers are invited to contribute!

4.2 Go

Bestiaries, anomalous positions and ruling for wizards (bent four in the corner and friends) could be looked at.

4.3 ToDo

- XiStrat examples for the standard problem categories (such as Grimshaw, Nowotny, Plachutta, Wurzburg-Plachutta, Bristol, Indian problem, Turton, Loyd-Turton, Brunner-Turton, Zepler-Turton, etc.)
- defining mathematical terms dealing with those problems

Chapter 5

Retrograde Analysis

5.1 Chess

This approach (see references) delivers a final analysis of all positions starting from decided entities such as mates (for ’all about six men’ see for example [264]). The tabled results should then of course be understood in terms of general rules and principles.

5.2 Go

Same here.

Chapter 6

Engines

First of all let's point to Appendix C for further reference.

6.1 Non-cooperative Games

6.1.1 General Remarks

We aim at strong and fast engines. Perhaps the underlying mathematical structures can be helpful.

6.1.2 GCJ and JNI

At the moment the engine is implemented in the high-level procedural language Java. But there exist parts where bytecode might become slower (even with JIT) than native code. GCJ offers an alternative (see bin/gcj_auto_trial.sh) nowadays running with speed comparable to the standard VM (improvements to garbage collector, inlining etc. might provide enhancements in the future). Another way is to use JNI and implement the critical parts of the engines in the C language.

6.1.3 Chess

For now especially the Chess variant engine has been implemented to greater extent within XiStrat.

There exist a simple multi-player functionality and a more specialized version for the standard case of two opponents.

In the latter a (nominally) depth-limited alpha-beta negamax algorithm in fail-soft mode with iterative deepening, simple Zobrist key hashing, a transposition table and thereby simple 'best-first' move ordering is used (in failing hard a bound information wouldn't be achieved, couldn't be saved and later reused). The hashkey for the transposition table is of LONG type (and so 64-bit architectures are recommended), and we use the lowest 24 bits for addressing. New entries simply overwrite former ones (not regarding the height at all), this is the simplest solution. Since the number of legal positions exceeds the 64th power of two, in principle hash collisions can occur, but as long as we check that no impossible move is done, that's no problem (see [249]). The number of entries (the 24th power of two) in the transposition table should include the minimal tree (about square root of (branching factor raised to the depth-th power) that is), and on the other hand everthing should still fit into the available memory (heap space). Due to some odd / even effect we should perhaps restrict the reuse of hits to those with the same parity of depth. At the moment we reach (depending on the board and the actual position) search depths up to a dozen plies.

Of course, if one could sort perfectly, then no search at all would be needed anymore. The evaluation function for leaf nodes is fully programmable and does for now nothing else than applying some heuristic rule-of-thumb material counting (Static State Evaluation). Some standard tricks (quiescence, null move etc.) in this field of computer science will improve the performance in the future. So far it's still only a proof-of-concept engine, for example we simply apply the static evaluation regardless of the character of the final position and thus suffer from the horizon effect in its purest form. Instead one shouldn't stop at hot situations and use selective extensions until obvious captures are accounted for.

Standard features like the detection of threefold repetition of position (resulting in a draw no matter what the material advantage may be) are of course still missing. And to simplify things our rulings at the moment treat a stalemate the same as a checkmate.

While there are pathological cases where the opposite is true (for example see [184], in general a deeper search gives higher strength.

One should be aware of search inconsistencies (for example during search values in the transposition table change whereas they were assumed to be constant when justifying some tricky algorithm) as a theoretical possibility, but at the moment we just don't care.

6.1.4 Go

At the moment we have a simple random engine (in case of strength zero) and playing near the last stone of its opponent (for higher strength). If you lose too often against it, then your attitude towards the game Go is perhaps a little bit problematical; please keep in mind it is a game to enjoy :-)

See [212] and [209] for aspects of combinatoric game theory.

Before trying to produce a complete engine, perhaps it's a good idea to concentrate on single aspects of the game when starting a computational treatment.

For example capturing races (semeai) can be treated by the following formula: An n point nakade shape filled by m opponent stones is equivalent to $(n^2 - 3n)/2 + 3 - m$ outside liberties. Other topics could be recognizing secure territories, life and death, patterns, endgame and so on.

Besides the standard treatment compares the nodes by a single value, and thereby different aspects are valued against each other but actually aren't comparable. Partial order evaluation is a way to take this into account.

Recent advances in the field of Go engine programming are based on Monte-Carlo methods.

At the moment Go programs can't win against professionals whilst in Chess machines can compete with the world champion. So it is often said that Go is much harder (whilst having simpler rules) than Chess. But actually the bigger search space is achieved by a larger board, and there are some rare cases where some rulesets differ, so definately the rules are not that easily falling from heaven! And the complexity class for Go (EXPTIME-complete, see Chapter 19) is only achieved when taking the Ko rule into account, whereas in Chess you don't need this (but one could use such a rule as well of course). Perhaps the most important problem for a Go engine is the evaluation function. In Chess one can focus on the material balance and use extensions for hot situations thus achieving a decent algorithm already (regarding speed while still making sense). In Go on the other hand, estimating secure territory or the status of a group of stones are perhaps not that easy. We'll see!

6.1.5 Reversi, Othello

To be done.

6.2 Puzzles Autosolving

Some sliding puzzles (for example see [165]) have a close natural connection to monoids and permutations.

Solving Loyd's 15-16 puzzle (and its variants) seems to rely on stabilizer chains, with 3 generators for the submonoid of all words stabilizing the gap in its final position. The puzzle defined by a monoid representation is reduced to a permutation group element factorization problem in a low-degree representation of the Schützenberger group (see [162] and [59] for the latter).

Also for Rubik game variants some stabilizer chains and other methods from CGT (computational group theory) could be applied here.

According to [67] up to now no methods have been found for solving problems like finding the diameter of the Cayley graph of the Rubik's cube group on its natural set of six generators (this is the same as the maximum number of quarter twists to restore the pristine state starting from an arbitrary position) that are significantly better than a brute-force depth first search.

Recent progress [158] shows that the minimum number of moves (this time using the face turn metric) to solve the Rubik's cube is 20, 21 or 22.

6.3 ToDo

- we have severe, quite reproducable problems with search inconsistencies in the Chess engine, with apparently increased frequency since iterative deepening feature has been added
- standard tricks, experiment with various algorithms (bitboard representation, PVS, sophisticated move ordering, search extensions, futility and null-move pruning, quiescence searches, aspiration)
- multi-threaded Chess engine (for multi-core machines)
- engines for Go, Reversi, n-in-a-row etc.
- JNI
- display the complete sequence of moves what the Chess engine is thinking about at the moment
- using group-theoretical data structures (perhaps using the GAP Java binding)
- let the engine think also on the opponent's time (pondering)
- general time management, that is to say the engine has got a given amount of time, and then the engine decides how deep it should think. this is in contrast to the current way to specify some strength as a static parameter. iterative deepening is the tool of choice then.
- bughouse
- 4 players with fixed alliances: 0,1 vs 2,3 or 0,2 vs 1,3, how to tell the engine to be in alliance (not changeable during a match)? This is quite different from psychological numP > 2 without alliances.
- more profiling
- communicate a draw, stalemate or resign
- infinite-sized boards

Part II

Mathematics Background

In this part scientific background information is provided. Be aware that the following chapters have not yet been proofread as thoroughly as they should. They are for courageous readers only.

Well yes, it seems our dear project has to do with about all other branches of mathematics one can think of, but a little bit of hands waving is necessary as well. We aren't mathematical crackpots, are we?

Perhaps some things here might turn out as mere mess. Since we are non-experts in the fields addressed, some inaccuracies are of course unavoidable. Anyway it is crucial to be able to explore the unknown terrain by using computation (that's how the saying goes).

But there is no reason for a too submissive behavior either. On one hand one can try to use information from other parts of science for our concrete strategy games. And on the other side in return, one might even ask if by the help of computational trial / error and learning by doing we address some interesting mathematics valuable in its own right. General theorems emerge starting from shared properties of a wealth of examples.

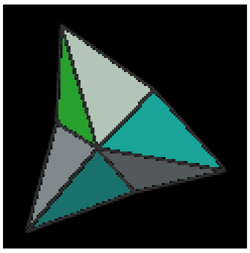
Chapter 7

Case Studies

Here various graphs are investigated. The classification is sometimes arbitrary, since a graph may arise as a result of different constructions (stellation, reticulation, truncation, glueing etc.). Some occuring mathematical terms will be explained later in subsequent chapters.

7.1 Tetrahedron Stellation I (Deltoid)

This graph arises as the stellation of a tetrahedron and is also known as a deltoid.



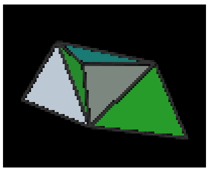
some parts are not visible in this view

Figure 7.1: *Stellated Tetrahedron*

vertices: 8, *faces:* 12, *edges:* 18, *genus:* 0
characteristic polynomial:
dual's characteristic polynomial:
chromatic number: 3
lattice group: A4 × S3

7.2 Truncated Deltoid

One part has been removed from the deltoid.



something was truncated

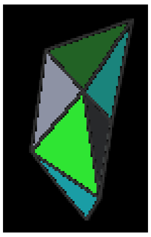
Figure 7.2: *Truncated Deltoid*

vertices: 7, *faces:* 10, *edges:* 15, *genus:* 0
characteristic polynomial:
dual's characteristic polynomial:

chromatic number: 3
lattice group (lossy): $Z(2) \times Z(3)^{10} \times A(10)$

7.3 Tetrahedron Iterated Stellation 3_1

Some more tetrahedra have been added at various locations.



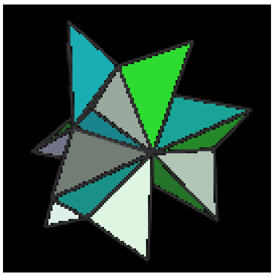
more iterations are imaginable

Figure 7.3: *Tetrahedron Stellation I(iterated)*

vertices: 7, *faces:* 10, *edges:* 15, *genus:* 0
characteristic polynomial:
dual's characteristic polynomial:
chromatic number: 3
lattice group (lossy): $Z(2)^{15} \times A(15)$

7.4 Tetrahedron Stellation II

The fully stellated tetrahedron has iteratively been stellated again.



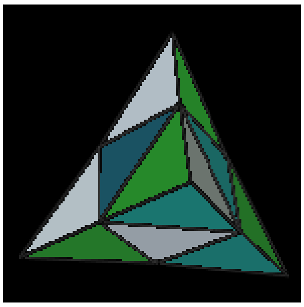
tetrahedron stellation iterated

Figure 7.4: *Tetrahedron Stellation II*

vertices: 20, *faces:* 36, *edges:* 54, *genus:* 0
characteristic polynomial:
dual's characteristic polynomial:
chromatic number: 3
lattice group: ((((C3 × ((C3 × C3) : C2)) : C2) : C3) : C2) × A4

7.5 Tetrahedron Reticulated

A subdivision of this sort if called reticulation.



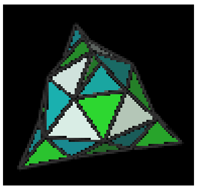
also called octahedron stellation I

Figure 7.5: *Tetrahedron Reticulated 0*

vertices: 10, *faces:* 16, *edges:* 24, *genus:* 0
characteristic polynomial: (2 + x)^2 (-6 - 4x + x^2)^3 (-2 + x^2)^3
dual's characteristic polynomial: (-3 + x) (-1 + x)^3 (1 + x)^6 (-5 + x^2)^3
chromatic number: 3
lattice group: $((C4 \times C4) : C3) : C2$

7.6 Tetrahedron Reticulated (big)

By this example you can see the principle of the construction more clearly now.



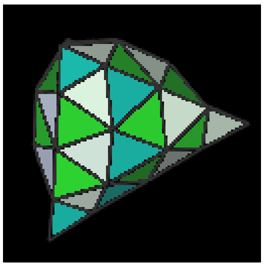
a little bigger than the previous graph

Figure 7.6: *Tetrahedron Reticulated I*

vertices: 20, *faces:* 36, *edges:* 54, *genus:* 0
characteristic polynomial:
dual's characteristic polynomial:
chromatic number: 3
lattice group: $((C6 \times C6) : C3) : C2$

7.7 Tetrahedron Reticulated (tall)

Once started, it's hard to stop the procedure again ...



even bigger

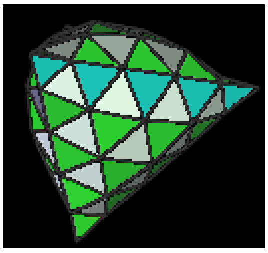
Figure 7.7: *Tetrahedron Reticulated II*

vertices: 34, *faces:* 64, *edges:* 96, *genus:* 0

characteristic polynomial:
dual's characteristic polynomial:
chromatic number: 3
lattice group: ((C8 $\times$ C8) : C3) : C2

7.8 Tetrahedron Reticulated (huge)

Of course we can even go one step further in this series of construction (and so we did).



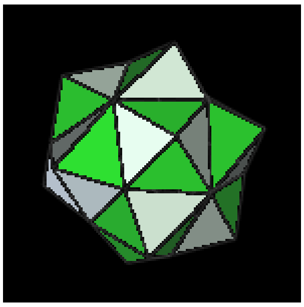
bigger than tall

Figure 7.8: *Tetrahedron Reticulated III*

vertices: 52, *faces:* 100, *edges:* 150, *genus:* 0
characteristic polynomial:
dual's characteristic polynomial:
chromatic number: 3
lattice group: ((C10 $\times$ C10) : C3) : C2

7.9 Octahedron stellation II

The standard fundamental regions for the Coxeter reflection group 43.



reflection group 43

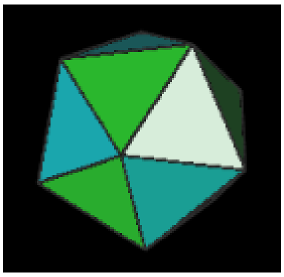
Figure 7.9: *Octahedron stellation II*

vertices: 26, *faces:* 48, *edges:* 72, *genus:* 0
characteristic polynomial: $(-4 + x)^3\ x^4\ (2 + x)^6\ (-2 + x^2)^5\ (-48 - 26x + x^3)$
dual's characteristic polynomial: $(-3 + x)\ (-2 + x)^2\ (-1 + x)^4\ x^4\ (1 + x)^4\ (2 + x)^2\ (3 + x)\ (-2 - 2x + x^2)^3\ (-2 + 2x + x^2)^3\ (2 - 4x - x^2 + x^3)^3\ (-2 - 4x + x^2 + x^3)^3$
chromatic number: 2
lattice group (lossy): $Z(2)^{15} \times Z(3)^4$

7.10 Icosahedron

7.10.1 Standard (regular)

See [4] for more information about the icosahedron (and other polyhedra).



good old

Figure 7.10: *Icosahedron*

vertices: 12, *faces:* 20, *edges:* 30, *genus:* 0

characteristic polynomial: (-5 + x)(-5 + x^2)^3 (1 + x)^5

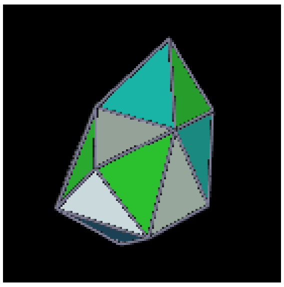
dual's characteristic polynomial: (-3 + x)(-1 + x)^5 x^4 (2 + x)^4 (-5 + x^2)^3

chromatic number: 3

lattice group: A(5)

7.10.2 Variant 1

After taking two neighboured faces and rotating them (generalization of a Rubik-like move).



a deformed icosahedron by rotating an area of two faces

Figure 7.11: *Icosahedron Variant 1*

vertices: 12, *faces:* 20, *edges:* 30, *genus:* 0

characteristic polynomial: (1 + x) (2 + x) (-1 - 2x + x^2) (-7 - 4x + 2x^2 + x^3) (22 + 43x + 9x^2 - 14x^3 - 3x^4 + x^5)

dual's characteristic polynomial: (-3 + x) (-1 + x)^3 x^2 (1 + x) (2 + x)^2 (-5 + x^2) (2 - 5x^2 + x^4) (-2 + 18x - 5x^2 - 9x^3 + x^4 + x^5)

chromatic number: 3

lattice group (lossy): $Z(2)^{29} \times A(30)$

7.11 Icosahedron Stellation I

A full stellation of the icosahedron (by tetrahedra).

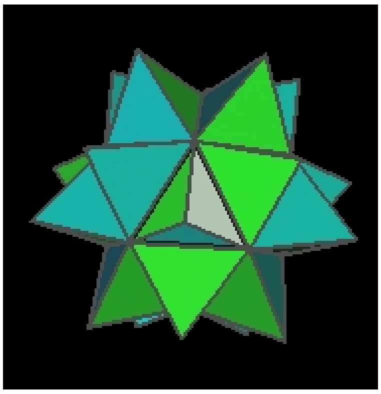
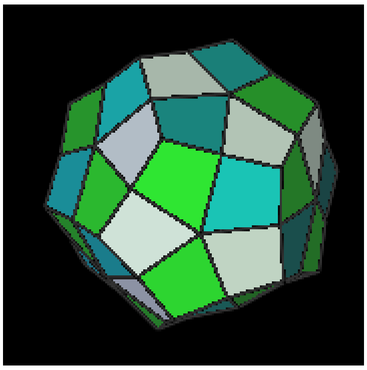


Figure 7.12: *Icosahedron Stellation I*

vertices: 32, *faces:* 60, *edges:* 90, *genus:* 0
characteristic polynomial: $x^8 (-15 - 5x + x^2) (-3 + x + x^2)^5 (5 - 20x - 15x^2 + x^4)^3$
dual's characteristic polynomial: $(-3 + x) x^{10} (2 + x)^{11} (-4 - x + x^2)^5 (-3 - x + x^2)^4 (-1 - x + x^2)^4 (4 + 6x - 5x^2 -2x^3 + x^4)^3$
chromatic number: 3
lattice group: $((A5 \times A5 \times A5) : C3) : C2$

7.12 Icosahedron Stellation II

Actually this graph was the result of a subsequent second stellation of the stellated icosahedron, but now when you look at it, it could also be called a reticulated dodecahedron.



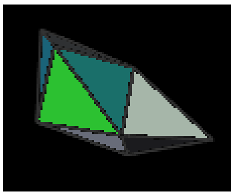
stellation with quads

Figure 7.13: *Icosahedron Stellation II*

vertices: 62, *faces:* 60, *edges:* 120, *genus:* 0
characteristic polynomial: $x^{12} (-4 + x) (-1 + x)^4 (1 + x)^4 (4 + x) (-2 - 2x + x^5) (-3 + x^2)^4 (-2 + 2x + x^2)^5 (44 - 16x^2 + x^4)^3$
dual's characteristic polynomial: $(-4 + x)(-1 + x)^4 x^6 (1 + x)^4 (5 - 5x + x^2)^3 (-5 + x^2)^4 (1 + 3x + x^2)^8 (4 - 7x - x^2 + x^3)^5$
chromatic number: 3
lattice group (lossy): $Z(2)\hat{}3 \times A(5)^4$

7.13 Cube Triang II

This is somehow derived from a triangulation of a cube, and turns out to be as well the glue between a 5-bipyramid and a tetrahedron.



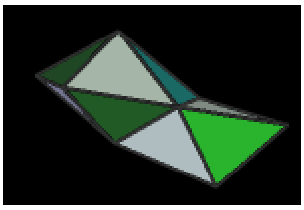
some glue of tetraheda results in just another simple group

Figure 7.14: *Cube Triang II*

vertices: 8, *faces:* 12, *edges:* 18, *genus:* 0
characteristic polynomial: (-1 + x + x^2) (-5 + 23x + 30x^2 - 11x^3 - 16x^4 - x^5 + x^6)
dual's characteristic polynomial: (-3 + x) x^2 (-4 - 9x - 2x^2 + 3x^3 + x^4) (-4 + 11x + x^2 - 7x^3 + x^5)
chromatic number: 3
lattice group : A(36)

7.14 Yabi

Yet another bipyramid based variation.



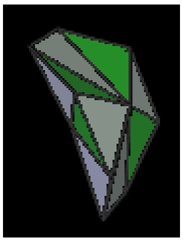
another bipyramid-based graph

Figure 7.15: *Yabi*

vertices: 9, *faces:* 14, *edges:* 21, *genus:* 0
characteristic polynomial:
dual's characteristic polynomial:
chromatic number: 3
lattice group (lossy): $Z(2) \times Z(3) \times A(14)^3$

7.15 Bipyramid Octa

This is also a bipyramid. Btw. it seems that bipyramids are often called lens spaces.



a bipyramid (rendered nontrivially)

Figure 7.16: *Bipyramid Octa*

vertices: 10, *faces:* 16, *edges:* 24, *genus:* 0
characteristic polynomial:
dual's characteristic polynomial:
chromatic number: 2
lattice group (lossy): $(((C4 \times C2) : C4) : C3) : C2$

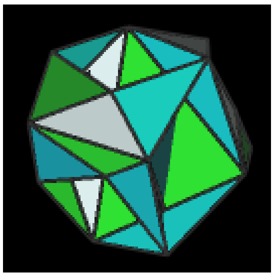
7.16 L2(7)

7.16.1 Cartographic

This is one possible construction on eight labels (a minimal faithful permutation representation would actually only need seven labels), but thereby an triangle has got an edge glued to a neighbouring edge (so now the dual_rot around the vertex there fixes an edge label), and since poly=3 as a side-effect this results in a loop for the remaining edge, which is perfectly valid but conflicts a little bit with the way we draw edges as straight lines (we could program something for this circumstances, draw some circle or so).
vertices: 5, *faces:* 4+1, *edges:* 8, *genus:* 0
characteristic polynomial: x $(x^4 - x^3 - 12x^2 - 6x + 8)$
dual's characteristic polynomial: $(x - 1)$ x $(x^2 - x - 4)$
chromatic number: 2
lattice group (lossy): L2(7) ≅ L3(2) ≅ GL3(2)

7.16.2 Regular

See [51] and [48] for more information about this regular representation using the period triple (2,3,7) and orbit genus 3. The dual graph has got the same group of course. And again a basic map together with some additional identification of boundaries could be more meaningful (Klein's configuration in [109]). Actually we could avoid self-intersections by allowing triangles with non-straight lines (see [113] and [112]). And Klein's quartic is involved as well. It's an extension of the concept of the Platonic solids to a hyperbolic heptagonal tiling.



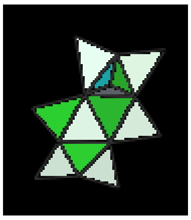
the simple group L2(7) of order 168 (hyperbolic geometry) with some self-intersections

Figure 7.17: *L2(7) Regular*

vertices: 24, *faces:* 56, *edges:* 84, *genus:* 3
characteristic polynomial: $(x - 7)\ (x + 7)\ (x^2 - 7)^8$
dual's characteristic polynomial: $(x - 3)\ x^7\ (x + 2)^6\ (x^2 - 2x - 1)^8\ (x^2 - 2)^6\ (x^2 + x - 4)^7$
chromatic number: 3
lattice group (lossy): L2(7) ≅ L3(2) ≅ GL3(2)

7.17 M(24)

See for example [48] for more information about this representation on 24 labels with passport (2^12, 3^6 1^6, 21^1 3^1) . The cartographic groups and Grothendieck's dessins d'enfants perfectly fit into our framework.



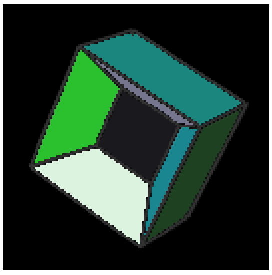
the sporadic simple Mathieu group M(24)

Figure 7.18: *M(24)*

vertices: 13, *faces:* 12+1, *edges:* 24, *genus:* 0
characteristic polynomial: x (6 + 118x + 48x^2 + 512x^3 + 29x^4 - 694x^5 - 273x^6 + 276x^7 + 160x^8 - 26x^9 - 24x^{10} + x^{12})
dual's characteristic polynomial: (-2 + x) x^2 (1 + x) (-1 + x + x^2)^2 (4 + 4x - 6x^2 - x^3 + x^4)
chromatic number: 3
lattice group (using the 'double'): M(24)

7.18 Torus 3x4

A quite trivial example, but our layouting procedure stubbornly insisted in a crumpled version with self-intersections, so for the embedding we figured out some reasonable coordinates by brute force.



(faked layout)

Figure 7.19: *Torus 3x4*

vertices: 12, *faces:* 12, *edges:* 24, *genus:* 1
characteristic polynomial: (x - 4) (x - 2)^2 (x - 1)^2 x (x + 1)^4 (x + 3)^2
dual's characteristic polynomial: (since this graph is self-dual, see above)
chromatic number: 3
lattice group : $(((C2 \times ((C3 \times C3) : C4)) : C2) : C2) : C2$

7.19 Trapezohedron

A polyhedron whose faces are trapeziums is called a trapezohedron. The trapezohedra are the dual polyhedra of the Archimedean antiprisms.

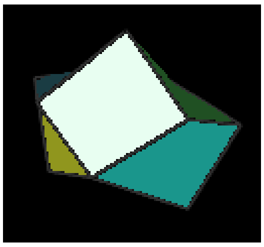
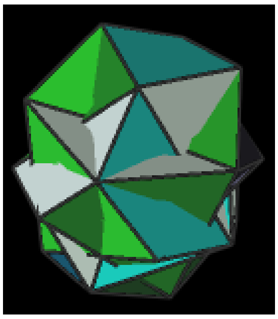


Figure 7.20: *Trapezohedron*

vertices: 10, *faces:* 8, *edges:* 16, *genus:* 0
characteristic polynomial:
dual's characteristic polynomial:
chromatic number: 4
lattice group (lossy): $Z(2)^{11} \times Z(3)$

7.20 S(5)

Again we use the regular representation of a small group. Is there some embedding without self-intersections as well? Probably.



S(5) as a compact Riemann surface of orbit genus 4 (with some self-intersections)

Figure 7.21: *S(5)*

vertices: 24, *faces:* 30, *edges:* 60, *genus:* 4

characteristic polynomial: (x - 5) (x - 1) (x + 1) (x + 5) $(x^2 - 5)^6$

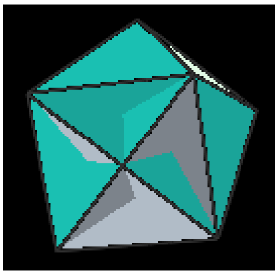
dual's characteristic polynomial: (x - 4) $(x - 2)^{11}$ x^5 $(x + 1)^4$ $(x + 2)^5$ $(x + 3)^4$

chromatic number: 3

lattice group: A(5) : Z(2) $\cong$ S(5)

7.21 A(6)

And again we give a fishy representation of a small simple group, with some self-intersections reminding us of the fact that this sort of display is rather limited. A map with indicated identifications of boundaries would be more appropriate this time probably.



A(6) as a compact Riemann surface of orbit genus 2

Figure 7.22: *A(6)*

vertices: 8, *faces:* 10, *edges:* 20, *genus:* 2

characteristic polynomial: (x - 5) (x - 1) $(x + 1)^2$ $(x^2 +2x - 1)^2$

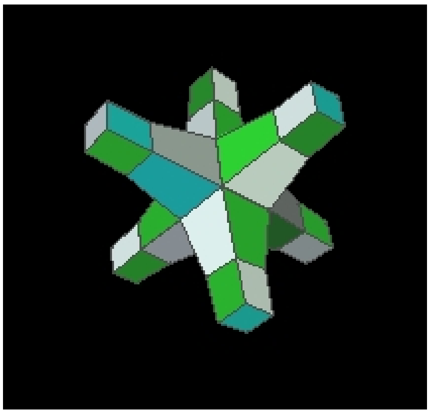
dual's characteristic polynomial: (x - 4) $(x - 1)^2$ (x + 1) $(x^2 + x - 4)$ $(x^2 + 2x - 1)^2$

chromatic number: 4

lattice group: A(6) $\cong$ L2(9)

7.22 Quad Star

A star-like object built from cubes.



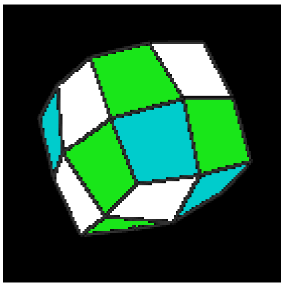
some cubes glued together

Figure 7.23: *Quad Star*

vertices: 56, *faces:* 54, *edges:* 108, *genus:* 0
characteristic polynomial: $(-3 + x)^2 (-1 + x)^{11} (1 + x)^{11} (3 + x)^2 (-3 + 12x - 7x^2 + x^3) (3 + 12x + 7x^2 + x^3) (-1 + 11x -x^2 - 4x^3 + x^4)^3 (-1 - 11x - x^2 + 4x^3 + x^4)^3$
dual's characteristic polynomial: $(-4 + x) (-3 - x + x^2) (-1 - x + x^2) (-1 - x + x^2)^3 (-1 + x + x^2)^3 (5 + 5x + x^2) (-3 - 6x + x^2 + 4x^3 + x^4)^3 (-8 + 5x + 14x^2 - 3x^3 - 4x^4 + x^5)^3 (18 + 23x - 8x^2 - 13x^3 + x^5)$
chromatic number: 3
lattice group (lossy): $Z(2)^{19} \times Z(3)^9 \times A(9)$

7.23 Reticulated 2x2 Cube

This is the lesser known cousin of the famous 3x3 instance.



a 2×2 Rubik's cube

Figure 7.24: *Cube 2x2*

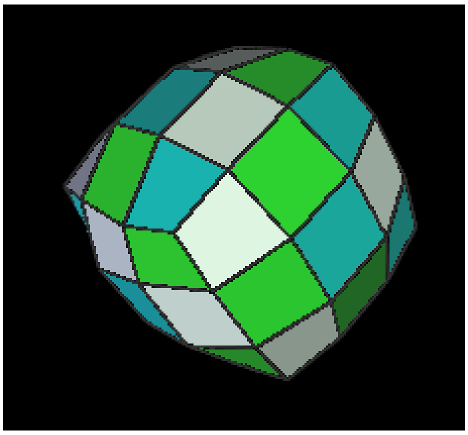
vertices: 26, *faces:* 24, *edges:* 48, *genus:* 0
characteristic polynomial:
dual's characteristic polynomial:
chromatic number: 3
lattice group (lossy): $Z(2)^{13} \times Z(3)^4$

7.24 Reticulated 3x3 Cube Variants

All the different graphs are variants arising out of the original graph by a morphing procedure.

7.24.1 standard Rubik's Cube

Not only puzzle kids are familiar with this graph. Our layout is not completely standard though, but somehow it resembles a ball.



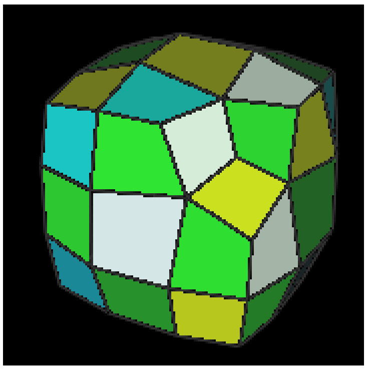
an (interestingly rendered) 3×3 Rubik's cube

Figure 7.25: *Cube 3x3*

vertices: 56, *faces:* 54, *edges:* 108, *genus:* 0
characteristic polynomial: $(1 - 3x + x^2)^2 (-1 - x + x^2)^6 (-1 + x + x^2)^6 (1 + 3x + x^2)^2 (6 - 5x - 3x^2 + x^3) (-6 - 5x + 3x^2 + x^3) (2 - 9x - x^2 + x^3)^3 (-2 - 9x + x^2 + x^3)^3$
dual's characteristic polynomial: $(-4 + x) (-2 + x) (2 + x) (-1 + x) x (1 + x) (-1 + 2x + x^2) (2 - 3x - 2x^2 + x^3) (-1 - 3x + x^2 + x^3) (18 + 9x - 13x^2 - x^3 + x^4) (12 + 111x - 306x^2 - 2002x^3 + 1484x^4 + 10231x^5 - 83x^6 - 19231x^7 - 5886x^8 + 14263x^9 + 5906x^{10} - 4936x^{11} - 2269x^{12} + 837x^{13} + 403x^{14} - 67x^{15} - 33x^{16} + 2x^{17} + x^{18})^2$
chromatic number: 3
lattice group (lossy): $Z(2)^{21} \times Z(3)^{11}$

7.24.2 rotated corner

You may guess (and your eyes/brain are actually not sure about it either) that there should be two allowed layouts for the corner, going outwards or into the inside. Compare the following polynomials with those of the original graph.



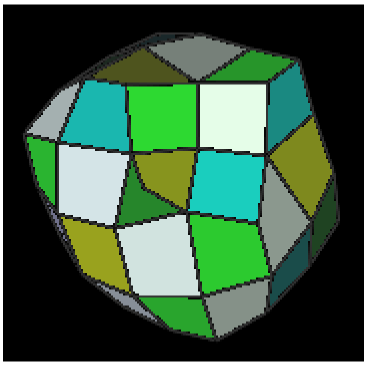
a 3×3 Rubik's cube with rotated corner

Figure 7.26: *3x3 rotated corner*

vertices: 56, *faces:* 54, *edges:* 108, *genus:* 0
characteristic polynomial: $x^2 (1 - 3x + x^2)^2 (-1 - x + x^2)^5 (-1 + x + x^2)^5 (1 + 3x + x^2)^2 (2 - 9x - x^2 + x^3)^2 (-2 - 9x + x^2 + x^3)^2 (-3244 + 14696x^2 - 22543x^4 + 14369x^6 - 4301x^8 + 619x^{10} - 41x^{12} + x^{14})$
dual's characteristic polynomial: $(-4 + x) (-2 + x) (2 + x) (-1 + x) x^3 (1 + x)^3 (-1 + 2x + x^2)^3 (2 - 3x - 2x^2 + x^3)^3 (-1 - 3x + x^2 + x^3)^3 (18 + 9x - 13x^2 - x^3 + x^4)^3 (2 - x - 9x^2 + x^3 + x^4)^2$
chromatic number: 4
lattice group (lossy): $Z(2)^2 \times Z(3)^{70} \times A(36)^2$

7.24.3 rotated 2x2 area

As you can see, here two faces are connected along two neighboured edges, but never mind (in what can follow, there arises even the possibility of a pentagon with 2 edges glued together thereby becoming a triangle).



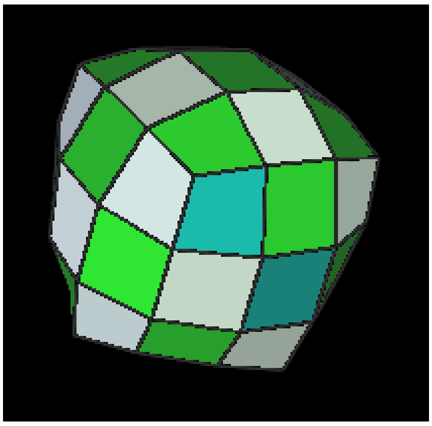
a 3×3 Rubik's cube with two faces connected via neighboured edges (after rotating 4 faces)

Figure 7.27: *3x3 rotated 2x2*

vertices: 56, *faces:* 54, *edges:* 108, *genus:* 0
characteristic polynomial:
dual's characteristic polynomial:
chromatic number: 4
lattice group (lossy): $Z(2)^2 \times A(108)^2$

7.24.4 rotated a 3x3 area

This is the result of rotating a 3×3 area (a 1/3 standard Rubik's cube move affecting 9 + 12 faces would give an isomorphic graph with the same polynomials and group).



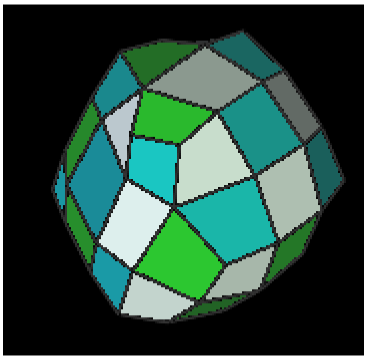
a Rubik's cube after rotating a 3×3 area

Figure 7.28: *rotated 3x3 area*

vertices: 56, *faces:* 54, *edges:* 108, *genus:* 0
characteristic polynomial: $(-1 - x + x^2)^2$ $(-1 + x + x^2)^2$ $(11 - 10x - 44x^2 + 36x^3 + 27x^4 - 12x^5 - 3x^6 + x^7)$ $(1 - 6x - 8x^2 + 28x^3 + 5x^4 - 12x^5 - x^6 + x^7)$ $(-1 - 6x + 8x^2 + 28x^3 - 5x^4 - 12x^5 + x^6 + x^7)$ $(-11 - 10x + 44x^2 + 36x^3 - 27x^4 - 12x^5 + 3x^6 + x^7)$ $(-1 + 97x^2 - 278x^4 + 143x^6 - 22x^8 + x^{10})^2$
dual's characteristic polynomial: $(-4 + x)$ $(-3 + x + 15x^2 + 4x^3 - 8x^4 - x^5 + x^6)$ $(-8 - 19x + 35x^2 + 47x^3 - 16x^4 - 16x^5 + x^6 + x^7)$ $(-2 - 7x + 13x^2 + 25x^3 - 6x^4 - 12x^5 + x^6 + x^7)$ $(-2 - 3x + 19x^2 + 9x^3 - 22x^4 - 8x^5 + 3x^6 + x^7)$ $(4 + 35x - 28x^2 - 387x^3 + 160x^4 + 971x^5 - 164x^6 - 714x^7 + 52x^8 + 210x^9 - 4x^{10} -25x^{11} + x^{13})^2$
chromatic number: 4
lattice group (lossy): $Z(2)^{53} \times A(27)$

7.24.5 a follow up to variant 3

After having rotated the top area counter-clockwise, then a (freshly formed) right-hand 3×3 area (not along a straight line though) has been moved one step counter-clockwise. Now how would you define new areas to be moved (more precisely: imagine rules to cut along edges)?



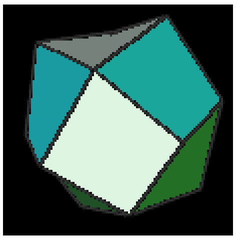
subsequently to variant 3, another move of a similar sort has been done

Figure 7.29: *3x3 follow up*

vertices: 56, *faces:* 54, *edges:* 108, *genus:* 0
characteristic polynomial: x^2 $(14 + 500x - 8834x^2 + 9004x^3 + 108312x^4 - 152919x^5 - 544248x^6 + 808010x^7 + 1459295x^8 - 2109449x^9 - 2299002x^{10} + 3068713x^{11} + 2223753x^{12} - 2625825x^{13} - 1353956x^{14} + 1385850x^{15} + 525805x^{16} - 464902x^{16} - 130168x^{18} + 100110x^{19} + 20181x^{20} - 13673x^{21} - 1877x^{22} + 1136x^{23} + 95x^{24} - 52x^{25} - 2x^{26} + x^{27})$ $(-14 + 500x + 8834x^2 + 9004x^3 - 108312x^4 - 152919x^5 + 544248x^6 + 808010x^7 - 1459295x^8 - 2109449x^9 + 2299002x^{10} + 3068713x^{11} - 2223753x^{12} - 2625825x^{13} + 1353956x^{14} + 1385850x^{15} - 525805x^{16} - 464902x^{16} + 130168x^{18} + 100110x^{19} - 20181x^{20} - 13673x^{21} + 1877x^{22} + 1136x^{23} - 95x^{24} - 52x^{25} + 2x^{26} + x^{27})$
dual's characteristic polynomial: $(-4 + x)$ $(56 + 540x - 198x^2 - ... + 2x^{25} + x^{26})$ $(112 + 896x - ... + 2x^{26} + x^{27})$
chromatic number: 3
lattice group (lossy): Z(2) × A(108)

7.25 Rhombic Dodecahedron

A dodecahedron always has got twelve faces, but strangely enough this time no pentagons are involved.



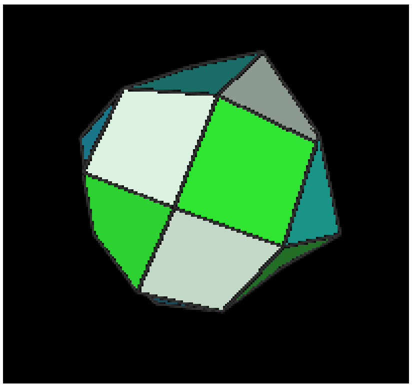
a zonotope

Figure 7.30: *Rhombic Dodecahedron*

vertices: 14, *faces:* 12, *edges:* 24, *genus:* 0
characteristic polynomial:
dual's characteristic polynomial:
chromatic number: 3
lattice group: (S4 × S4) : C2

7.26 Zonotope 5

A zonotope is the Minkowski sum of finitely many line segments. The linear image of an hypercube in n-space is also a possible definition.



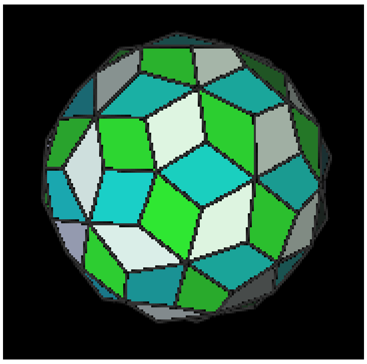
(with 5 generators)

Figure 7.31: *Zonotope 5*

vertices: 22, *faces:* 20, *edges:* 40, *genus:* 0
characteristic polynomial: $(-1 + x^5)$ $(1 + x^5)$ $(1 - 3x + x^2)^2$ $(1 + 3x + x^2)^2$ $(-10 - x + x^2)$ $(-10 + x + x^2)$
dual's characteristic polynomial: x $(-4 + x)$ $(-8 + x^2)$ $(-1 - 5x - 6x^2 + x^4)^2$ $(5 - 5x - 4x^2 + 2x^3 + x^4)^2$
chromatic number: 3
lattice group (lossy): $Z(2)^2 \times Z(5)^8 \times A(8)$

7.27 Ball Eneninda

Derived from a standard soccerball by triangulation of the pentagons and hexagons with a subsequent deflation of pairs.



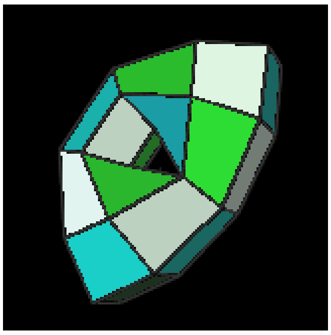
a ball

Figure 7.32: *Ball Eneninda*

vertices: 92, *faces:* 90, *edges:* 180, *genus:* 0
characteristic polynomial:
dual's characteristic polynomial:
chromatic number: 3
lattice group (lossy): $Z(2)^3 \times Z(3)^2 \times A(5)^6$

7.28 Monkgau

You might not want to know, but we actually intended to involve some Gaussian map of the monkey saddle this time. Since we got lost in our construction, this strange graph was the result. It also has a right to live, so we didn't eliminate it.



just another graph

Figure 7.33: *Monkgau*

vertices: 29, *faces:* 29, *edges:* 58, *genus:* 1

characteristic polynomial: (-1 + x^5) (1 + x^5) (1 - 3x + $x^2)^2$ (1 + 3x + $x^2)^2$ (-10 - x + x^2) (-10 + x + x^2)

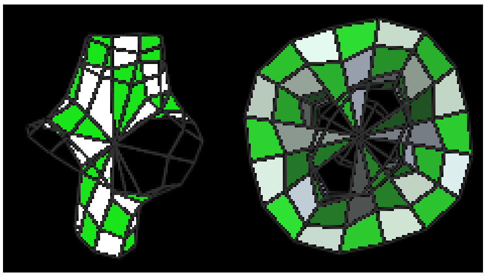
dual's characteristic polynomial: x (-4 + x) (-8 + x^2) (-1 - 5x - 6x^2 + $x^4)^2$ (5 - 5x - 4x^2 + 2x^3 + $x^4)^2$

chromatic number: 3

lattice group (lossy): $Z(2)^{58} \times A(58)$

7.29 Costa-Hoffman minimal surface

A completely immersed minimal surface with finite total curvature, boundary, genus 1 and 3 ends. It's also called a 3-punctured torus.



mimicking Weierstrass parametrization

Figure 7.34: *Costa-Hoffman*

vertices: 77, *faces:* 64+3, *edges:* 144, *genus:* 1

characteristic polynomial: (5 - 5x + x^2) ... (1 - 144x^2 + ... + $x^{16})^2$

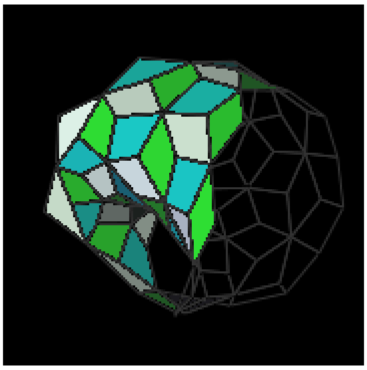
dual's characteristic polynomial:

chromatic number: 2

lattice group (lossy): $Z(2)^{75}$

7.30 Quad Hex tesselation on a torus

This somehow reminds of Escher's works.



this tesselation with boundary on a torus

Figure 7.35: *Quad Hex*

vertices: 86, *faces:* 72+1, *edges:* 159, *genus:* 1

characteristic polynomial: x^{24} (-270848 + 2736704x^2 - ... - 56x^{24} + x^{26}) (4356233216 - 50968580096x^2 + ... - 103x^{34} + x^{36})

dual's characteristic polynomial: x (2 + x)^{18} (-1 + 9x + 127x^2 - ... - 13x^{22} + x^{23}) (-448 - 1624x + ... - 23x^{29} + x^{30})

chromatic number: 3

lattice group (using the double, lossy version): $Z(2) \times A(434)$

(labels pointing towards holes fixed during flip): $Z(2) \times A(288)$

(treating the hole as a normal face): $Z(2) \times A(159)^2$

7.31 Quasicrystals

These tilings are aperiodic. All examples presented here have been constructed using an inflation/deflation procedure on two types of rhombs. The examples given differ in the substitution rule used and in the initial configuration to start the iterations from.

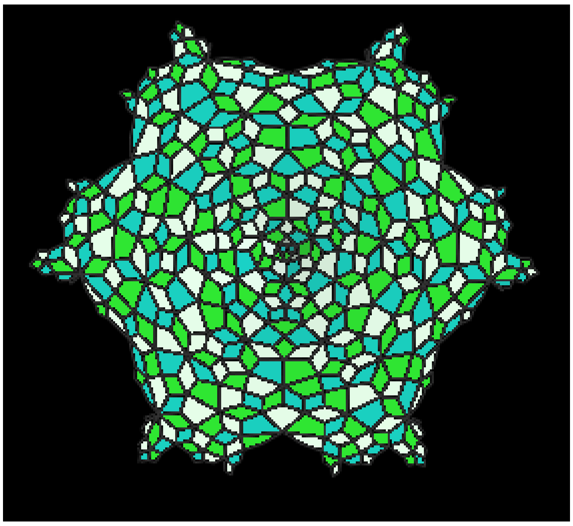
7.31.1 Penrose substitution

The following examples are due to Penrose. One remarkable property of the Penrose tilings is that every finite portion of any tiling is contained infinitely often in every other tiling. As a consequence, no finite patch of tiles determines the rest of the tiling, it is impossible to tell from any patch of tile which tiling it is on, and only at their infinite limits are the different patterns distinguishable. Btw. the golden ratio is involved as well.

7.31.1.1 Triagonal Penrose Quasicrystals

There are many graphs that can be constructed by the inflation/deflation procedure,

7.31.1.1.1 Iteration Four Here we only show this final iteration.



iteration four

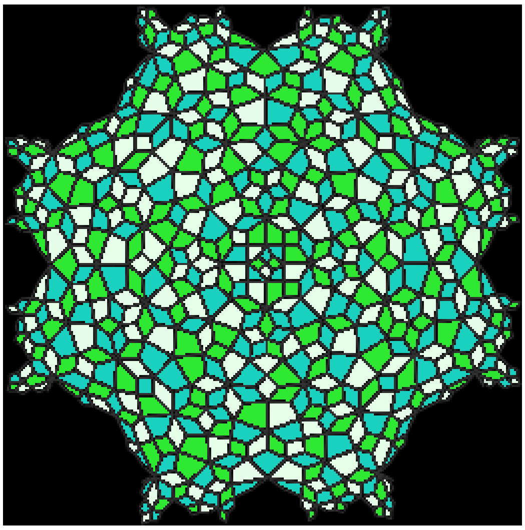
Figure 7.36: *Triagonal Penrose Quasicrystal IV*

vertices: 532, *faces:* 444+1, *edges:* 775, *genus:* 0
characteristic polynomial:
dual's characteristic polynomial:
chromatic number: 3
lattice group:

7.31.1.2 Quadrangular Penrose Quasicrystals

Starting with a four-fold symmetry,

7.31.1.2.1 Iteration Four We only show this final iteration.



iteration four

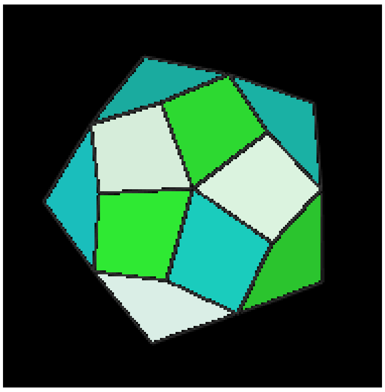
Figure 7.37: *Quadrangular Penrose Quasicrystal IV*

vertices: 709, *faces:* 592+1, *edges:* 1300, *genus:* 0
characteristic polynomial:
dual's characteristic polynomial:
chromatic number: 3
lattice group:

7.31.1.3 Pentagonal configuration

In this concrete example, the even variants are also called sun patterns, whereas the odd iterations represent the star patterns.

7.31.1.3.1 Initial decagon The traditional approach to start with.

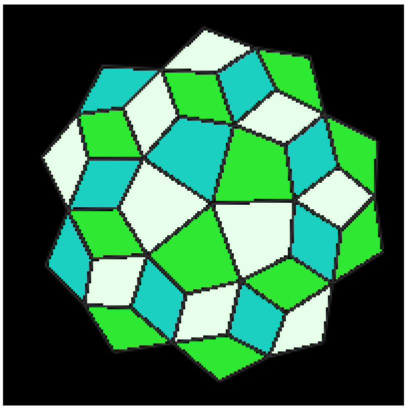


initial decagon

Figure 7.38: *Pentagonal Penrose Quasicrystal 0*

vertices: 16, *faces:* 10+1, *edges:* 25, *genus:* 0
characteristic polynomial: $x^4 (20 - 13x^2 + x^4) (4 - 6x^2 + x^4)^2$
dual's characteristic polynomial: $(-4 - 2x + x^2) (1 - 4x - 4x^2 + x^3 + x^4)^2$
chromatic number: 3
lattice group: $(C5 \times (((C5 \times ((C5 \times C5) : C2)) : C2) : C3)) : C2$

7.31.1.3.2 Iteration one Already more interesting.

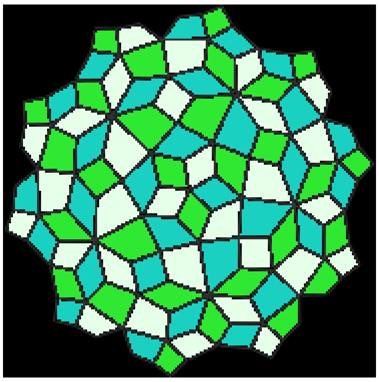


after one iteration

Figure 7.39: *Pentagonal Penrose Quasicrystal I*

vertices: 41, *faces:* 30+1, *edges:* 70, *genus:* 0
characteristic polynomial: $x^{11} (-1 + x)^4 (1 + x)^4 (-24 + 83x^2 - 20x^4 + x^6) (576 - 552x^2 + 179x^4 - 23x^6 + x^8)^2$
dual's characteristic polynomial: $(-1 + x + x^2) (8 + 8x - 5x^2 - 3x^3 + x^4) (-41 - 100x + 284x^2 + 694x^3 + 40x^4 - 661x^5 - 258x^6 + 218x^7 + 119x^8 - 27x^9 - 19x^{10} + x^{11} + x^{12})$
chromatic number: 3
lattice group (lossy): $Z(2) \times Z(5)^{15} \times A(15)$

7.31.1.3.3 Iteration two After another round of inflation and deflation.



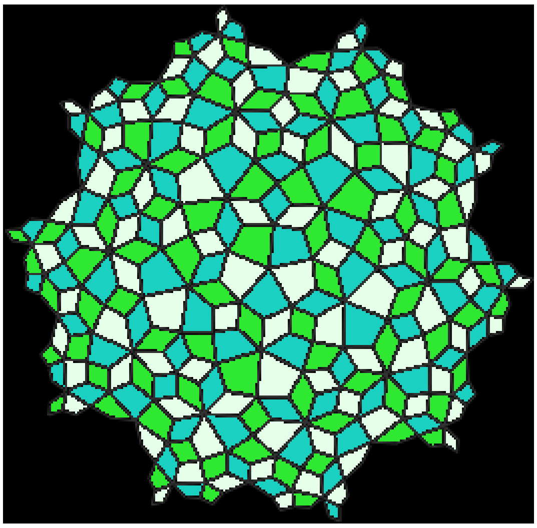
after two iterations

Figure 7.40: *Pentagonal Penrose Quasicrystal II*

vertices: 116, *faces:* 90+1, *edges:* 205, *genus:* 0
characteristic polynomial:
dual's characteristic polynomial:
chromatic number: 3
lattice group (lossy): $\mathrm{Z}(2)^{2} \times \mathrm{Z}(5)^{53} \times \mathrm{A}(53)$

7.31.1.3.4 Iteration three

The next iteration.



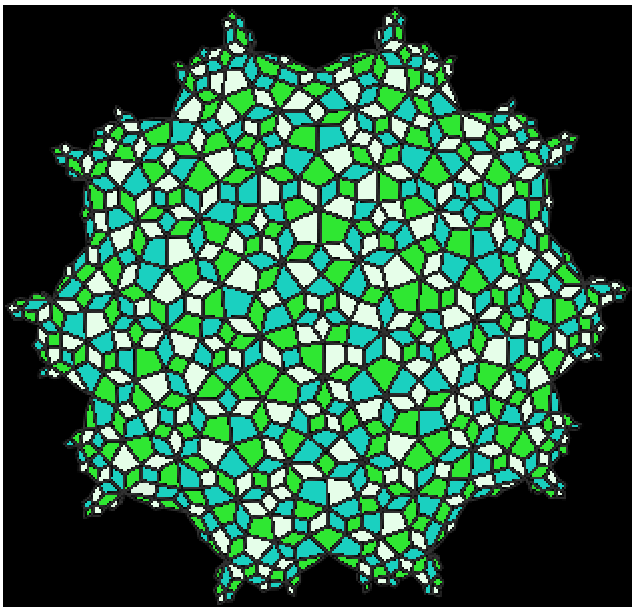
after three iterations (optimal are 3 colors)

Figure 7.41: *Pentagonal Penrose Quasicrystal III*

vertices: 321, *faces:* 260+1, *edges:* 580, *genus:* 0
characteristic polynomial:
dual's characteristic polynomial:
chromatic number: 3
lattice group (lossy):

7.31.1.3.5 Iteration four

The boundary exhibits some sort of self-similarity.



after four iterations (optimal are 3 colors)

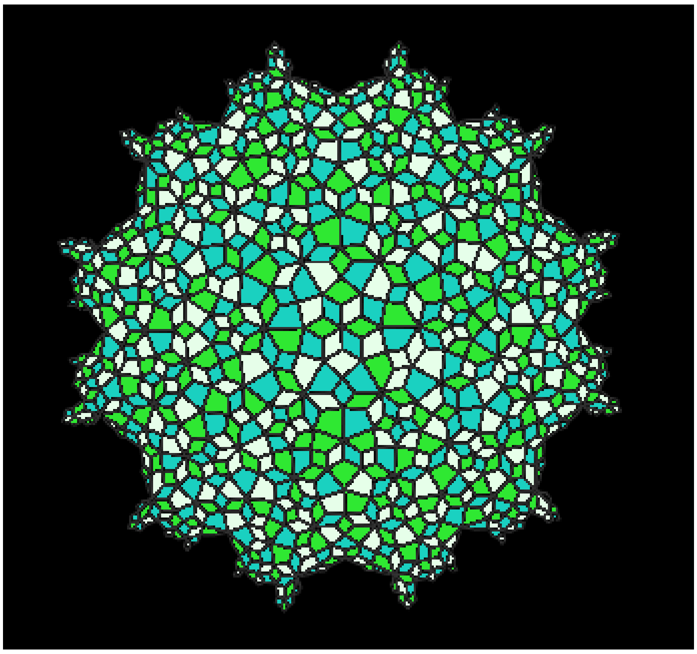
Figure 7.42: *Pentagonal Penrose Quasicrystal IV*

vertices: 886, *faces:* 740+1, *edges:* 1625, *genus:* 0
characteristic polynomial:
dual's characteristic polynomial:
chromatic number: 3
lattice group (lossy):

7.31.1.4 Hexagonal Penrose Quasicrystals

Starting with a six-fold symmetry,

7.31.1.4.1 Iteration Four We only show the fourth iteration.



iteration four

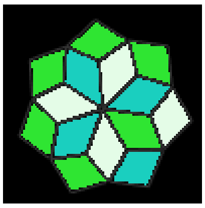
Figure 7.43: *Hexagonal Penrose Quasicrystal IV*

vertices: 1063, *faces:* 888+1, *edges:* 1950, *genus:* 0
characteristic polynomial:
dual's characteristic polynomial:
chromatic number: 3
lattice group:

7.31.1.5 Heptagonal Penrose Quasicrystals

There are still more graphs that can be constructed using an inflation/deflation procedure.

7.31.1.5.1 Initial 14-gon This is the initial configuration.

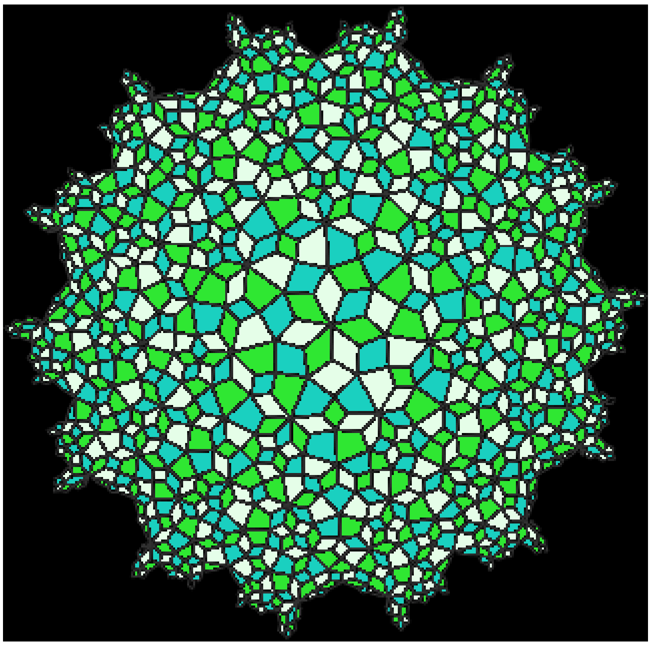


initial 14-gon

Figure 7.44: *Heptagonal Penrose Quasicrystal 0*

vertices: 22, *faces:* 14+1, *edges:* 35, *genus:* 0
characteristic polynomial:
dual's characteristic polynomial:
chromatic number: 3
lattice group: (C7 × (((C7 × ((C7 × C7) : C2)) : C2) : C3)) : C2

7.31.1.5.2 Iteration Four Here we show the fourth iteration.



iteration four

Figure 7.45: *Heptagonal Penrose Quasicrystal IV*

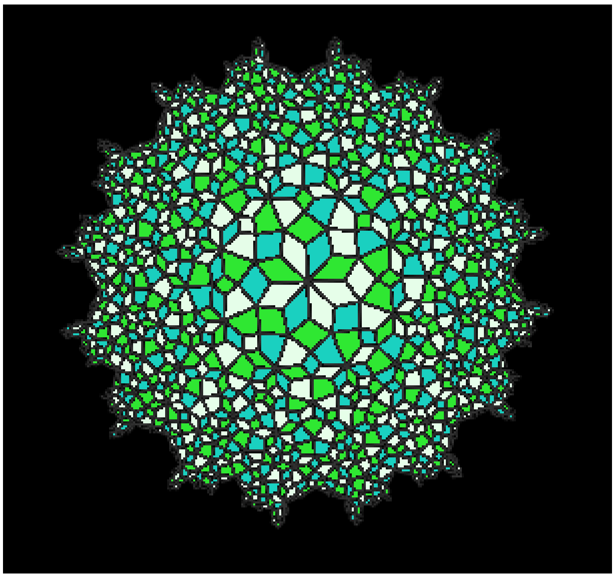
vertices: 1240, *faces:* 1036+1, *edges:* 1275, *genus:* 0
characteristic polynomial:
dual's characteristic polynomial:
chromatic number: 3

lattice group:

7.31.1.6 Octagonal Penrose Quasicrystals

Starting with a eight-fold symmetry,

7.31.1.6.1 Iteration Four We only show the fourth iteration.



iteration four

Figure 7.46: *Octagonal Penrose Quasicrystal IV*

vertices: 1417, *faces:* 1184+1, *edges:* 2599, *genus:* 0

characteristic polynomial:

dual's characteristic polynomial:

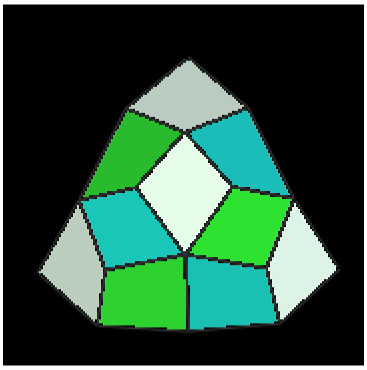
chromatic number: 3

lattice group:

7.31.1.7 Cartwheel Penrose Quasicrystals

In some ways the Cartwheel pattern is the most important Penrose tiling.

7.31.1.7.1 Initial decagon Originally the graph has only mirror symmetry. Later on one should not hesitate to apply a little trimming to see that (after an even number of iterations) the whole patch has 5-fold rotational symmetry (removing of boundary faces only connected the the graph over one single edge, thus making the layout insensitive against non-flat constellations, is in place).



initial decagon

Figure 7.47: *Cartwheel Penrose Quasicrystal 0*

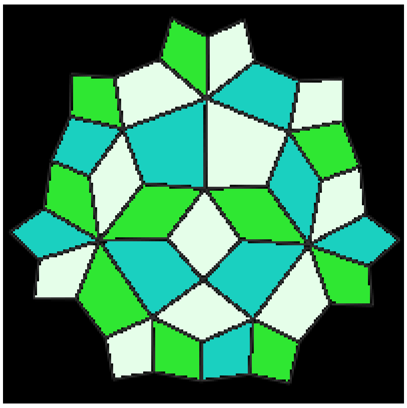
vertices: 16, *faces:* 10+1, *edges:* 25, *genus:* 0

characteristic polynomial:

dual's characteristic polynomial:

chromatic number: 3

lattice group: S(38)



iteration one

Figure 7.48: *Cartwheel Penrose Quasicrystal I*

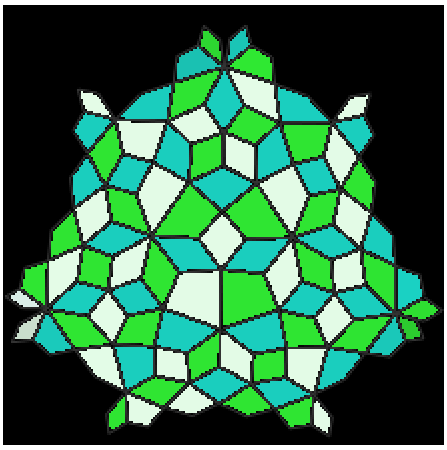
7.31.1.7.2 Iteration One *vertices:* 46, *faces:* 30+1, *edges:* 75, *genus:* 0

characteristic polynomial:

dual's characteristic polynomial:

chromatic number: 3

lattice group: S(100)



iteration two

Figure 7.49: *Cartwheel Penrose Quasicrystal II*

7.31.1.7.3 Iteration Two

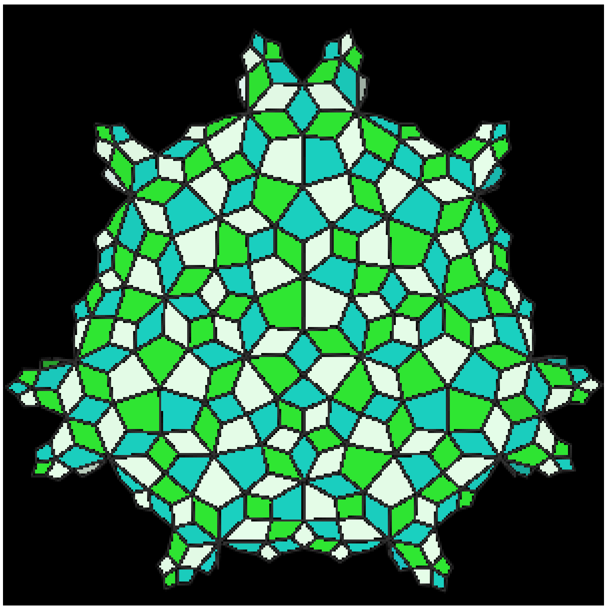
vertices: 126, *faces:* 90+1, *edges:* 215, *genus:* 0

characteristic polynomial:

dual's characteristic polynomial:

chromatic number: 3

lattice group:



iteration three

Figure 7.50: *Cartwheel Penrose Quasicrystal III*

7.31.1.7.4 Iteration Three

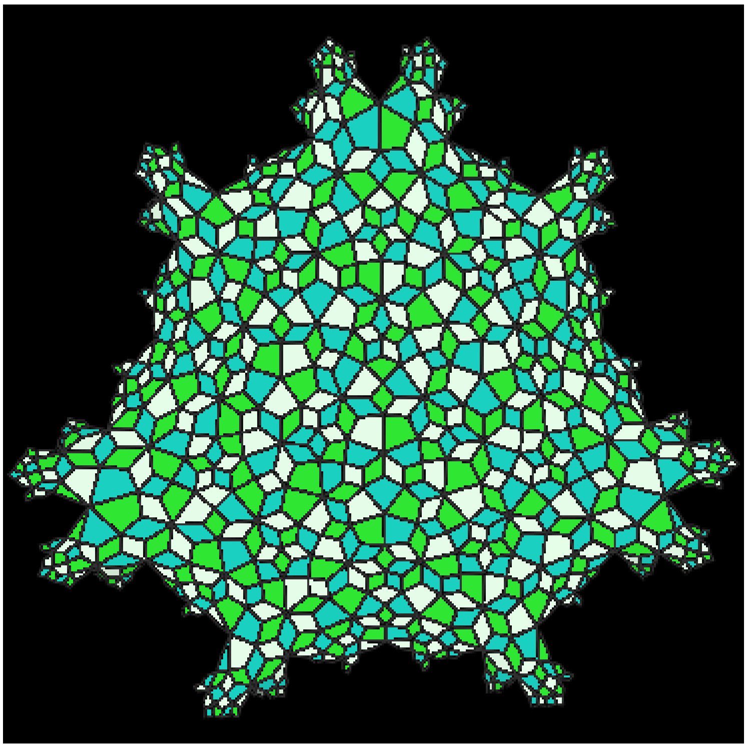
vertices: 356, *faces:* 270+1, *edges:* 625, *genus:* 0

characteristic polynomial:

dual's characteristic polynomial:

chromatic number: 3

lattice group:



iteration four

Figure 7.51: *Cartwheel Penrose Quasicrystal IV*

7.31.1.7.5 Iteration Four *vertices:* 996, *faces:* 790+1, *edges:* 1785, *genus:* 0

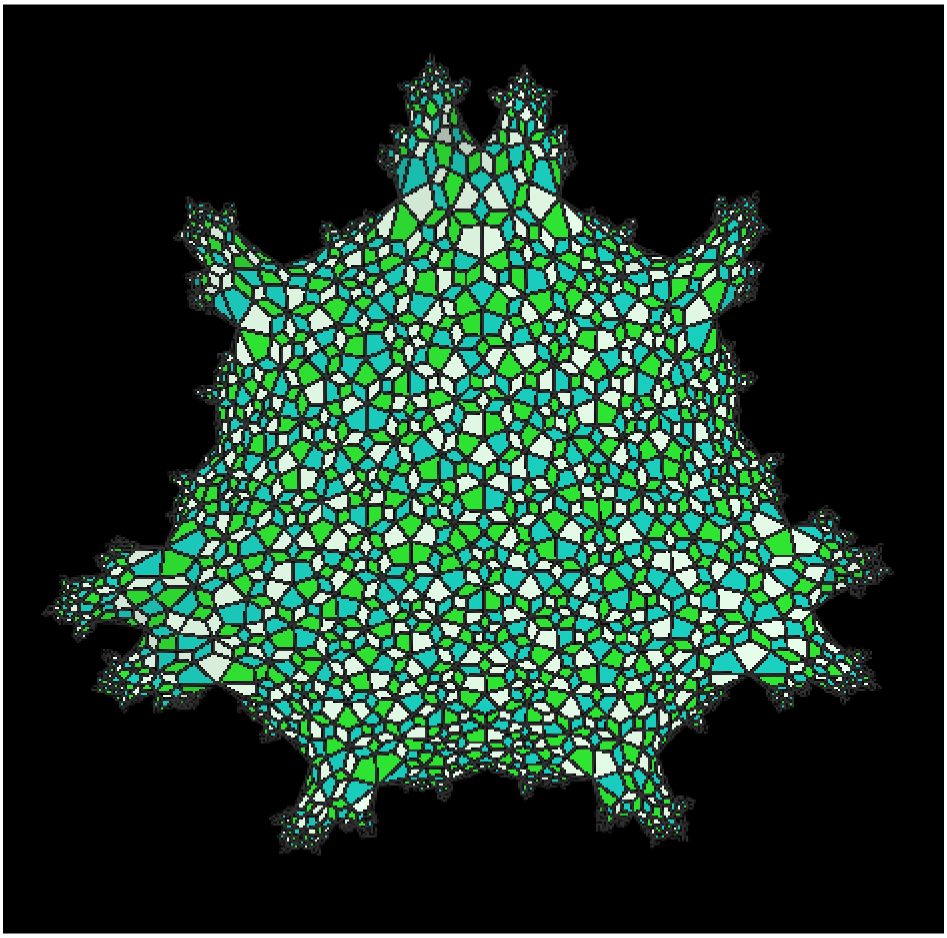
characteristic polynomial:

dual's characteristic polynomial:

chromatic number: 3

lattice group:

7.31.1.7.6 Iteration Five The outer portion of the pattern consists of two parts. There are ten sectors and ten spokes. Except for the original tiles at the center, every tile in a cartwheel tiling is contained in a patch of tiles that has symmetry group D5.



iteration five

Figure 7.52: *Cartwheel Penrose Quasicrystal V*

vertices: 2766, *faces:* 2270+1, *edges:* 5035, *genus:* 0

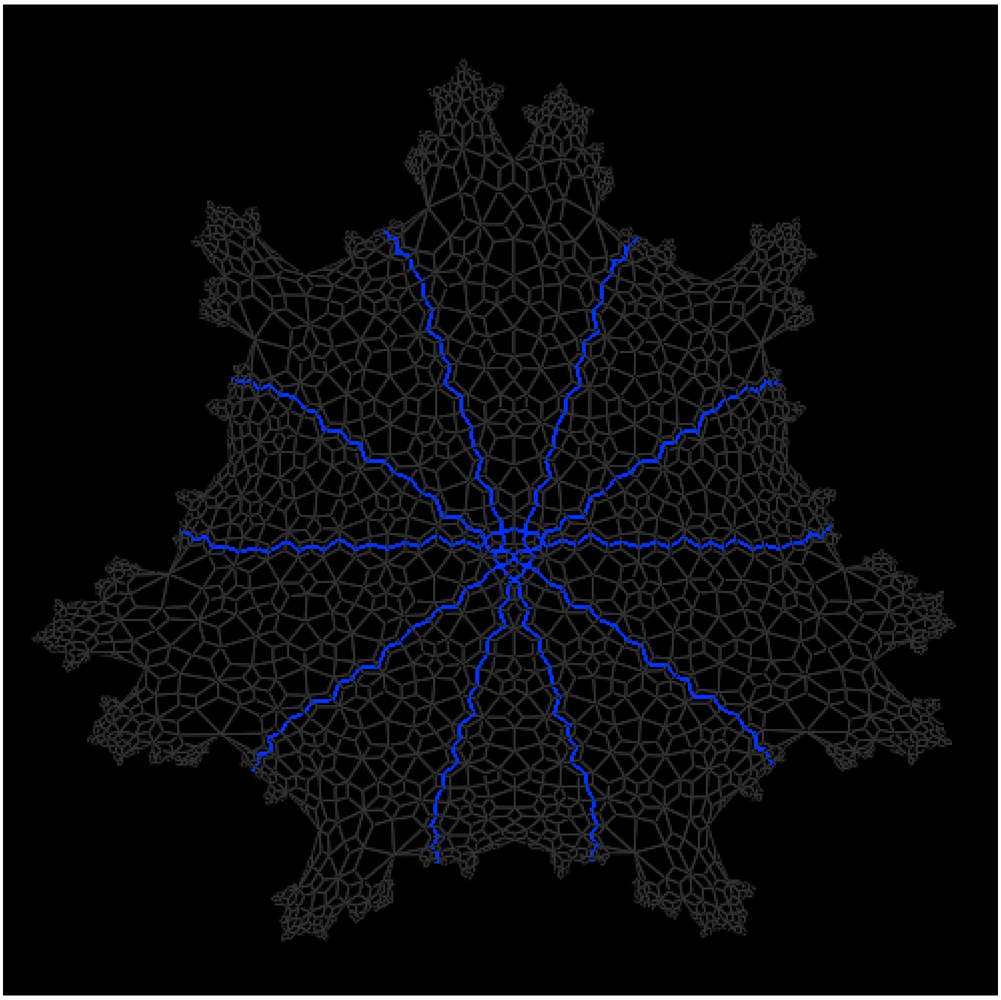
characteristic polynomial:

dual's characteristic polynomial:

chromatic number: 3

lattice group:

Here we show the ten (blue) spokes more closely.



lattice and spokes after iteration five

Figure 7.53: Cartwheel Penrose Quasicrystal V Spokes

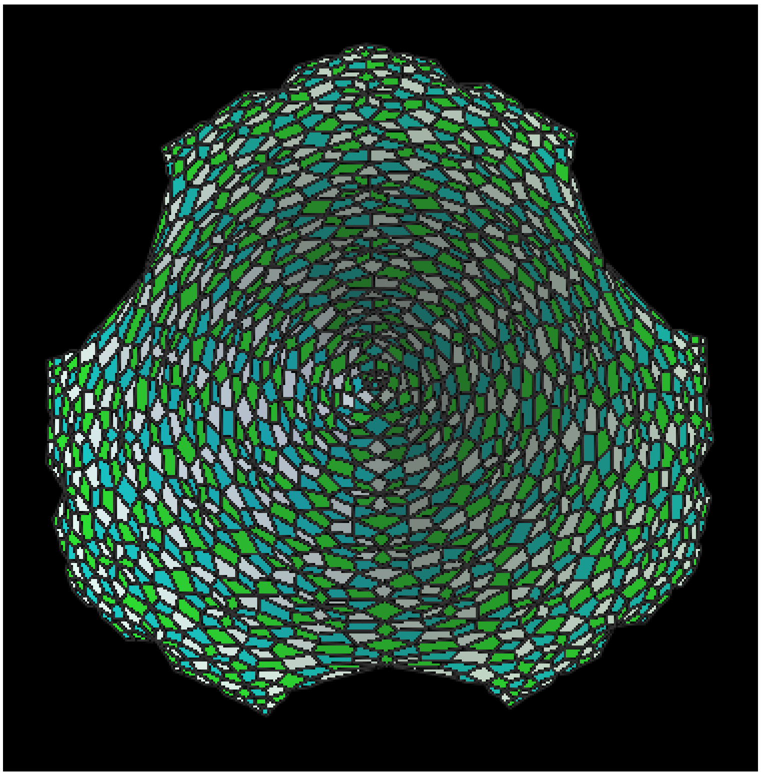
7.31.2 Ammann-Beenker substitution

The following examples are constructed using a substituion rule due to R. Ammann and F. Beenker. Btw. as in the Penrose case, one can construct a decoration by Ammann bars, and Conway worms are present as well. Different to the Penrose rulings, a local decoration forcing the local matching must be extended beyond the boundaries of the prototiles. In other words, we changed our whole quasicrystal procedure from reconstructing a suitable orientation in the follow-up step to the more efficient way of settling things directly in the precessing step during deflation. The initial configurations used are just the same as in the Penrose case above.

7.31.2.1 Triagonal Ammann-Beenker Quasicrystals

Again we use the inflation/deflation procedure and start with a three-fold symmetry.

7.31.2.1.1 Iteration Three Here we only show this final iteration, and since each iteration multiplies the number of faces to a larger extent than with the Penrose rule, we stop at the third iteration already. The result is far from being flat.



iteration three

Figure 7.54: *Triagonal Ammann-Beenker Quasicrystal III*

vertices: 1519, *faces:* 1437+1, *edges:* 2955, *genus:* 0

characteristic polynomial:

dual's characteristic polynomial:

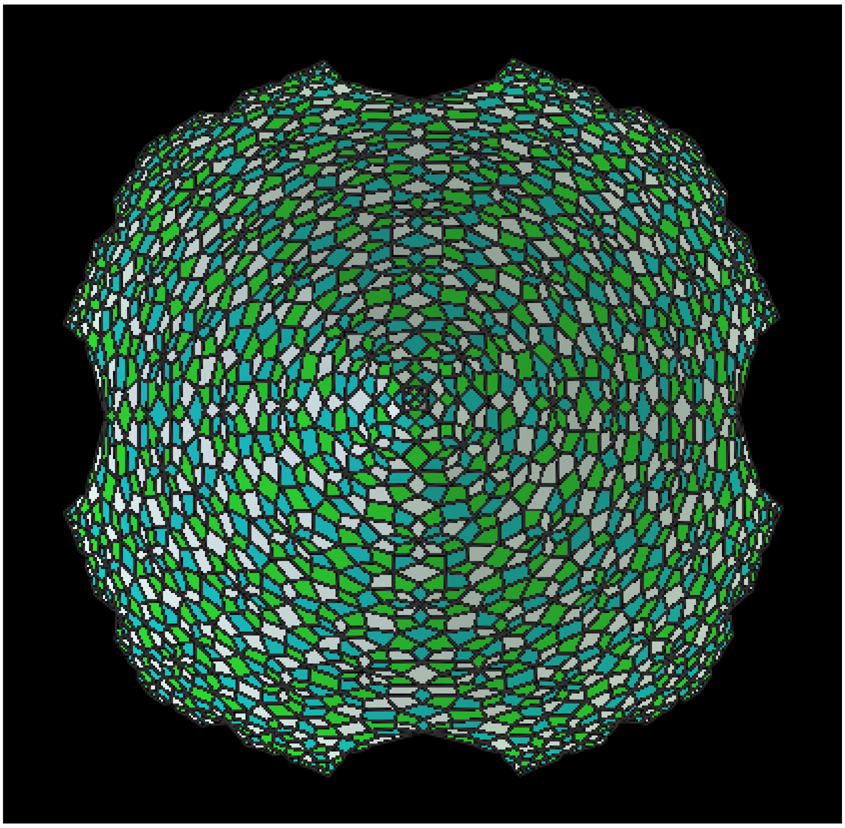
chromatic number: 3

lattice group:

7.31.2.2 Quadrangular Ammann-Beenker Quasicrystals

Starting with a four-fold symmetry,

7.31.2.2.1 Iteration Three We only show this final iteration.



iteration three

Figure 7.55: *Quadrangular Ammann-Beenker Quasicrystal III*

vertices: 2025, *faces:* 1916+1, *edges:* 3940, *genus:* 0

characteristic polynomial:

dual's characteristic polynomial:

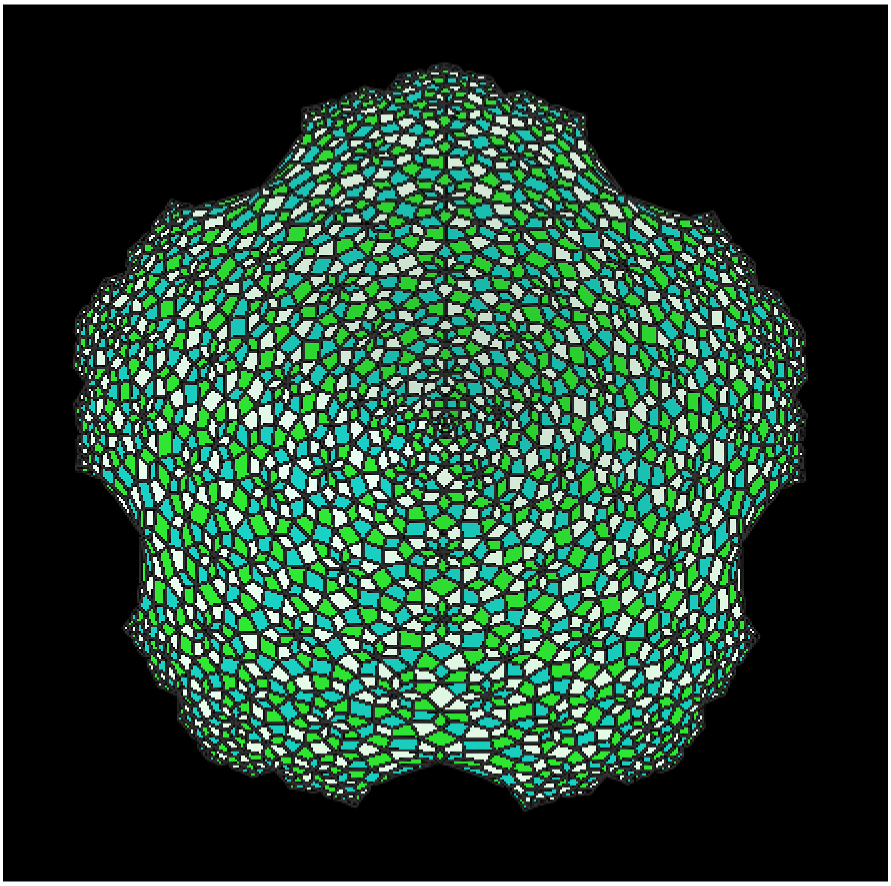
chromatic number: 3

lattice group:

7.31.2.3 Pentagonal configuration

Now the graphs become flatter already.

7.31.2.3.1 Iteration three We waited long enough until the layouting was settled.



after three iterations

Figure 7.56: *Pentagonal Ammann-Beenker Quasicrystal III*

vertices: 2531, *faces:* 2395+1, *edges:* 4925, *genus:* 0

characteristic polynomial:

dual's characteristic polynomial:

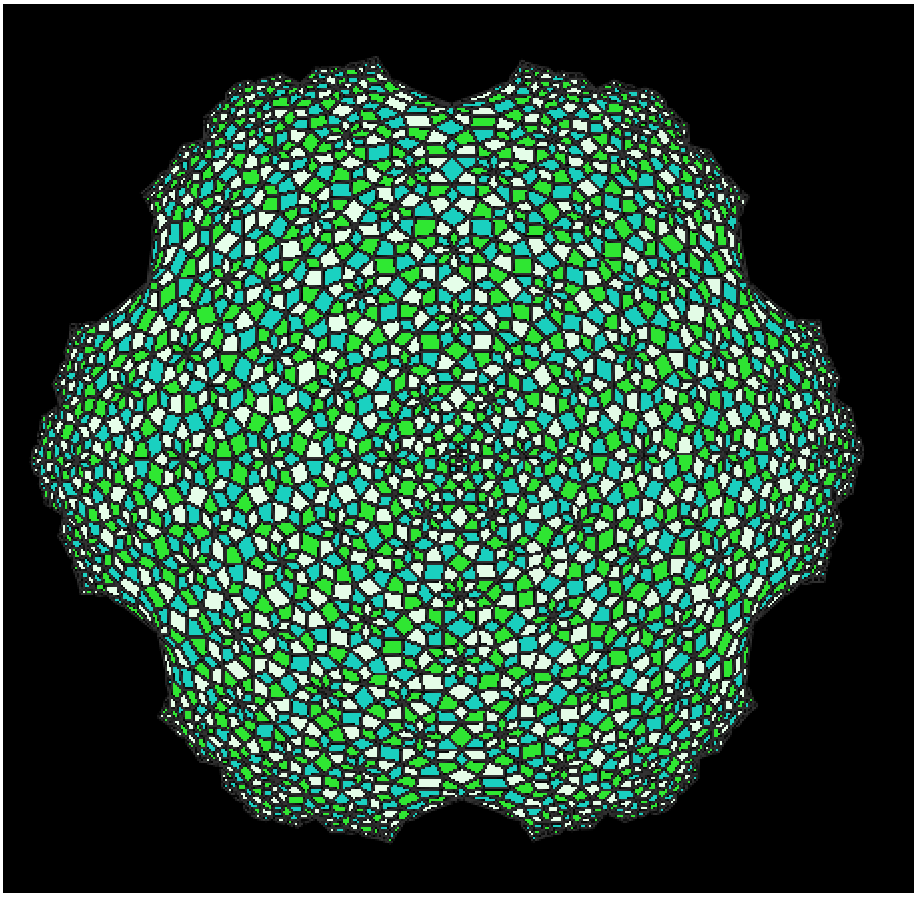
chromatic number: 3

lattice group (lossy):

7.31.2.4 Hexagonal Ammann-Beenker Quasicrystals

Starting with a six-fold symmetry,

7.31.2.4.1 Iteration Three We only show the third iteration.



iteration three

Figure 7.57: *Hexagonal Ammann-Beenker Quasicrystal III*

vertices: 3037, *faces:* 2874+1, *edges:* 5910, *genus:* 0

characteristic polynomial:

dual's characteristic polynomial:

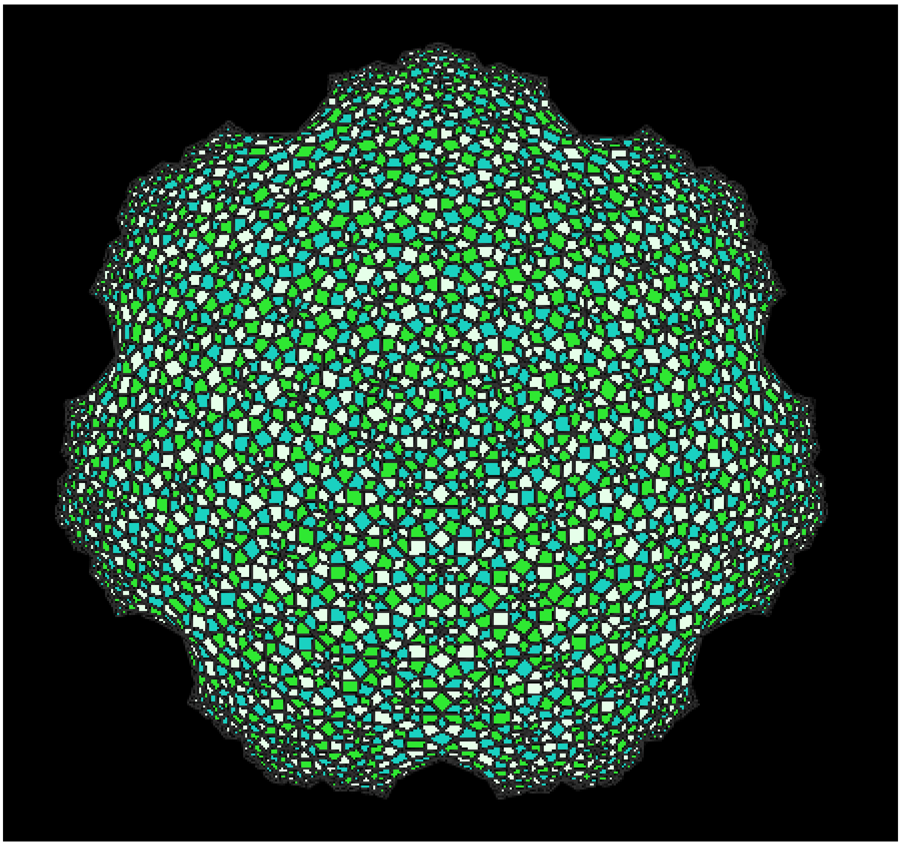
chromatic number: 3

lattice group:

7.31.2.5 Heptagonal Ammann-Beenker Quasicrystals

Completely flat graphs now.

7.31.2.5.1 Iteration Three Here we show the third iteration.



iteration three

Figure 7.58: *Heptagonal Ammann-Beenker Quasicrystal III*

vertices: 3543, *faces:* 3353+1, *edges:* 6895, *genus:* 0

characteristic polynomial:

dual's characteristic polynomial:

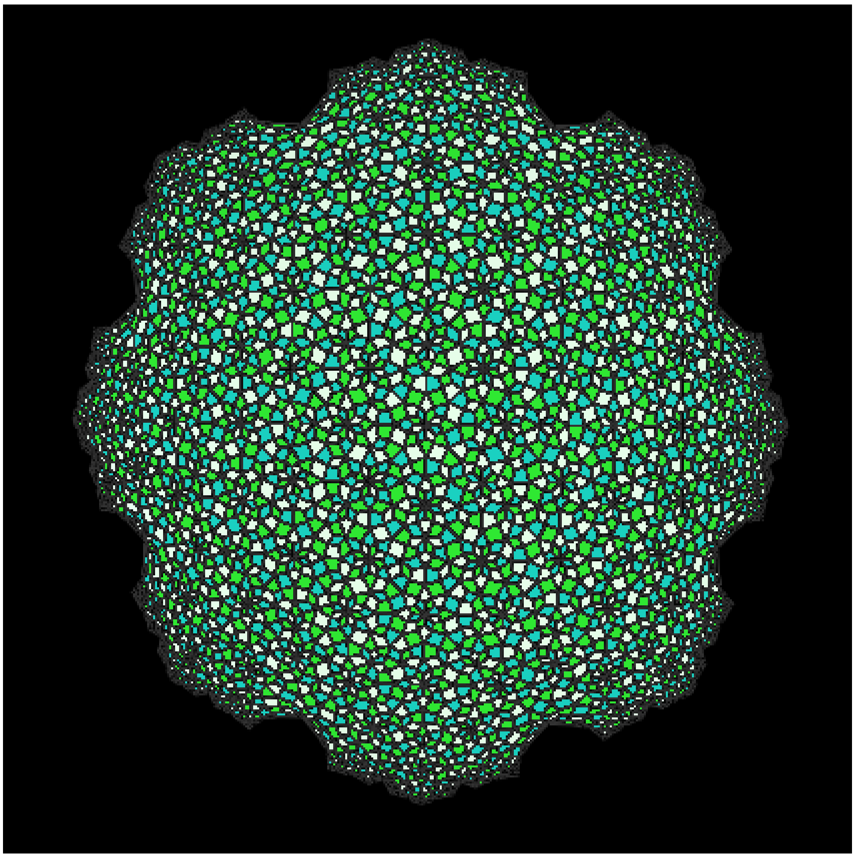
chromatic number: 3

lattice group:

7.31.2.6 Octagonal Ammann-Beenker Quasicrystals

Starting with a eight-fold symmetry. This is the traditional context where this substitution rule is used, probably because one can construct such a flat tesselation with two sorts of tiles of fixed angles and lengths. And here the so-called silver ratio (1+sqrt(2)) is involved.

7.31.2.6.1 Iteration Three We only show the third iteration. We didn't wait long enough for the layouting to reach a perfect state.



iteration three

Figure 7.59: *Octagonal Ammann-Beenker Quasicrystal III*

vertices: 4049, *faces:* 3832+1, *edges:* 7880, *genus:* 0

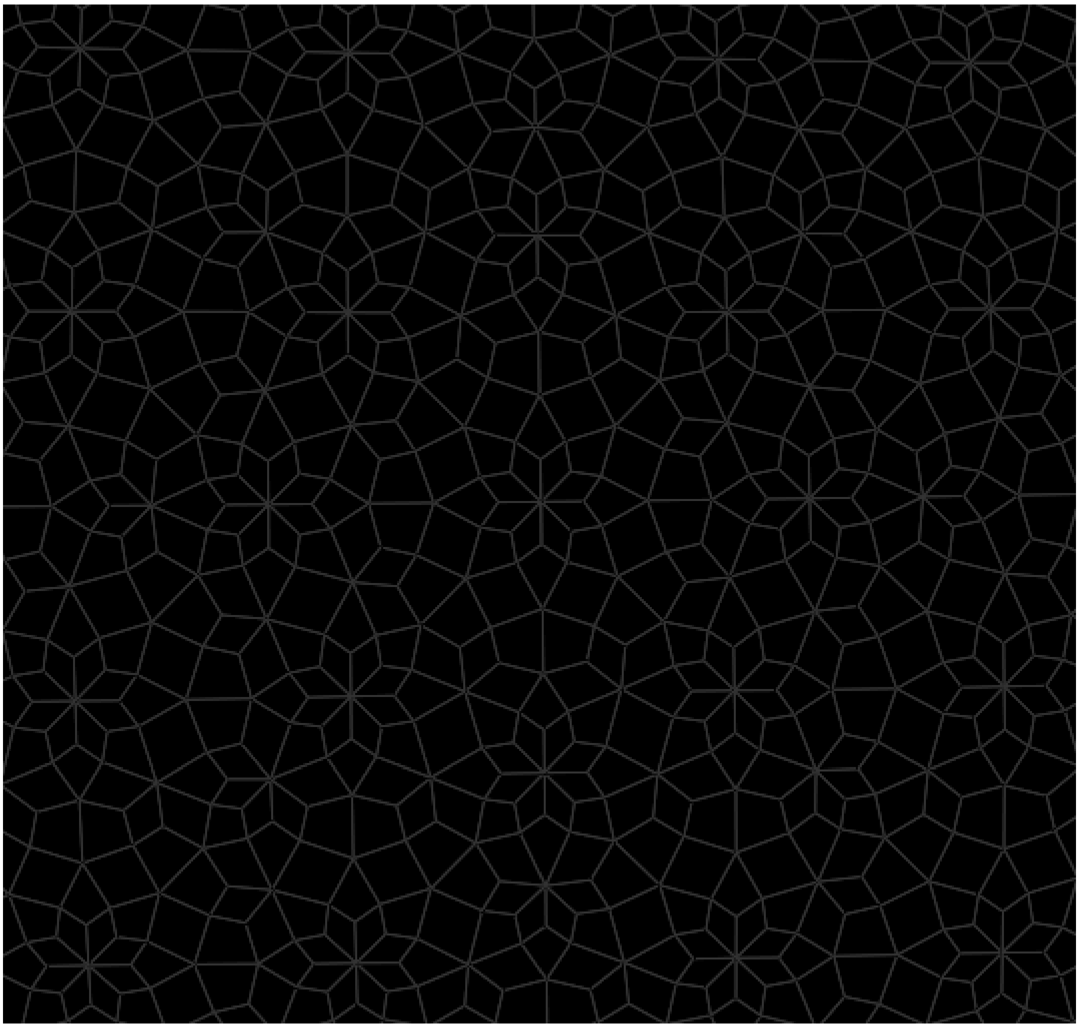
characteristic polynomial:

dual's characteristic polynomial:

chromatic number: 3

lattice group:

Here we show only the lattice grid. Pay attention to the fact that our layouting produces more than just two types of tiles (in fact these are not even rhombic but only quadrangular), but the eight-fold symmetry can be felt throughout.



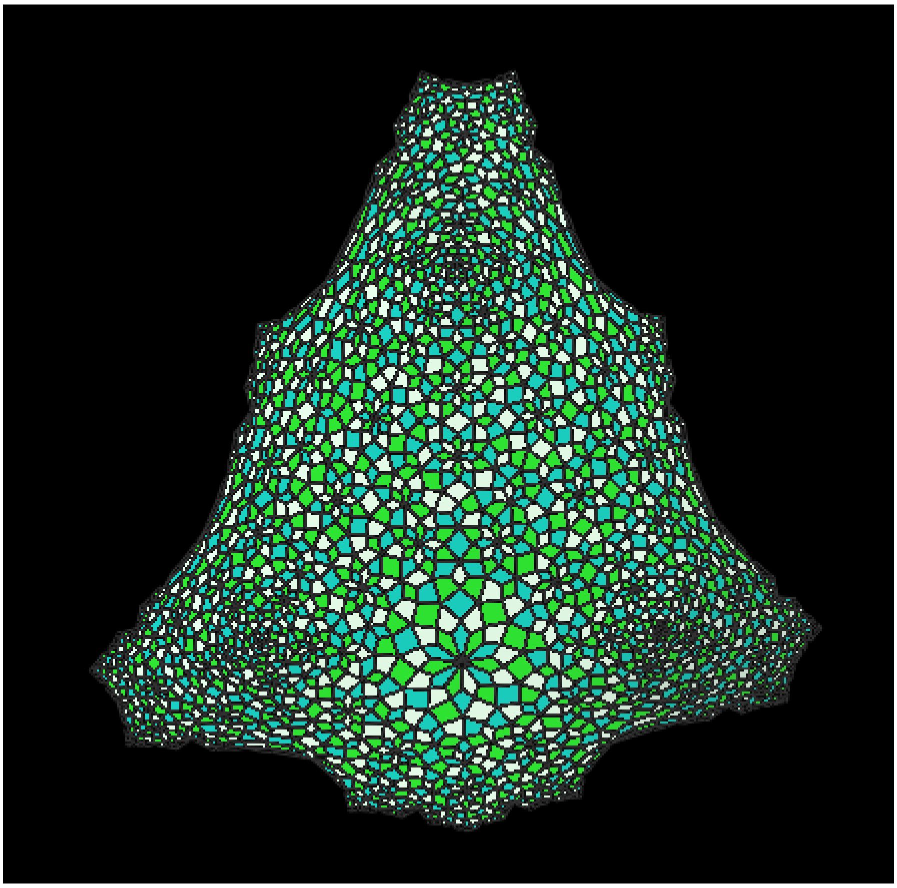
lattice after iteration three

Figure 7.60: Octagonal Ammann-Beenker Quasicrystal III Lattice

7.31.2.7 Cartwheel Ammann-Beenker Quasicrystals

There is no reason not to try this approach.

7.31.2.7.1 Iteration Three This looks quite interesting.



iteration three

Figure 7.61: *Cartwheel Ammann-Beenker Quasicrystal III*

vertices: 2531, *faces:* 2395+1, *edges:* 4925, *genus:* 0

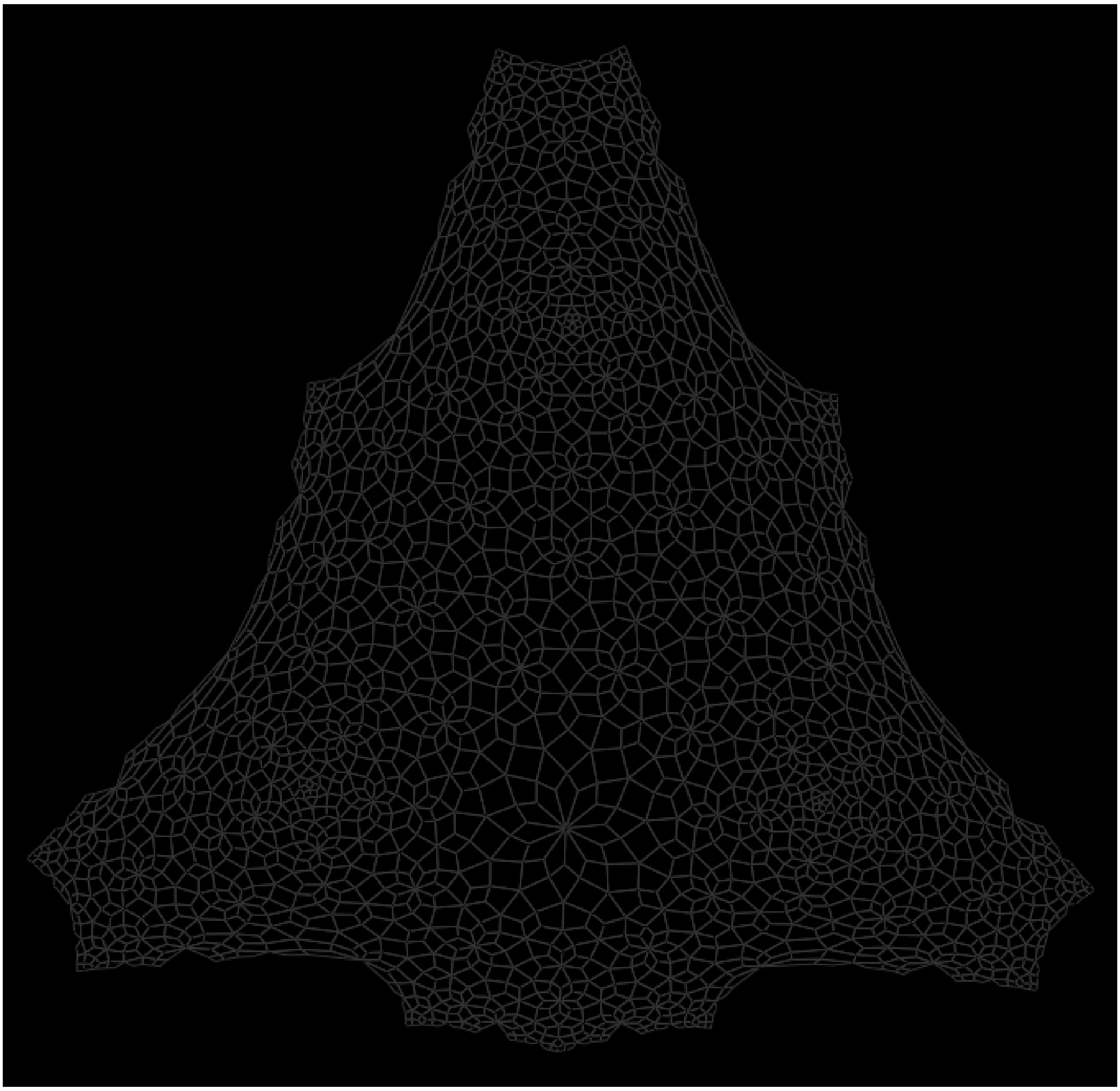
characteristic polynomial:

dual's characteristic polynomial:

chromatic number: 3

lattice group:

Since the random coloring obfuscates some of the details, we as well show the backside of the graph. The original configuration can still be depicted, and in the lower part an unexpected 11-fold symmetric pattern has emerged.



lattice after iteration three

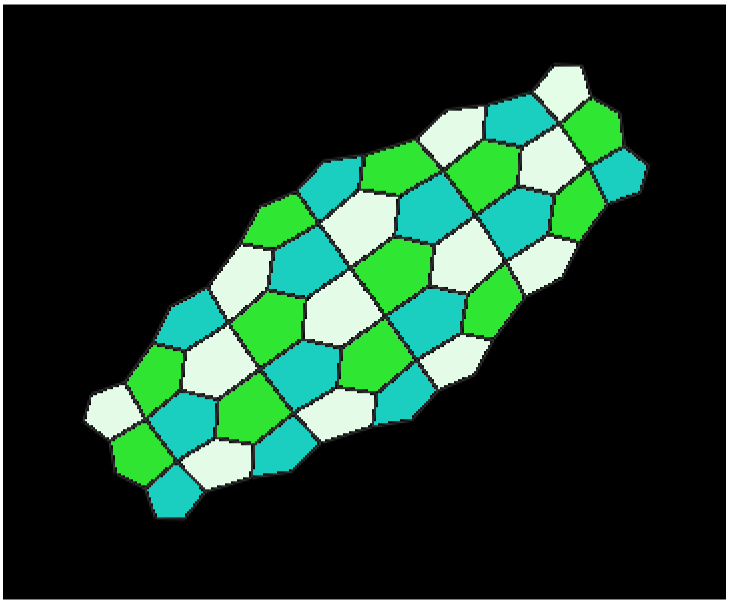
Figure 7.62: *Cartwheel Ammann-Beenker Quasicrystal III Lattice*

7.32 Pentagonal tilings of the plane

Of course those polygons must be irregular. We took some monohedral (and duohedral) standard tilings and then started our layouting. It is understood that the resulting graphs are somewhat different from the configurations we started from in that sometimes multiple sorts of tiles are present afterwards. And tilings differing by angles or lengths may turn out to lead to the same tiling in our framework. Besides the actual graphs are of minor interest anyway (extend the tiling, the graph changes, as does the group, the number of faces etc.), the dual space is what we are really targetting at.

7.32.1 Pentagonal Prismatic

This periodic tiling can as well be achieved by starting with a tiling due to Kershner.



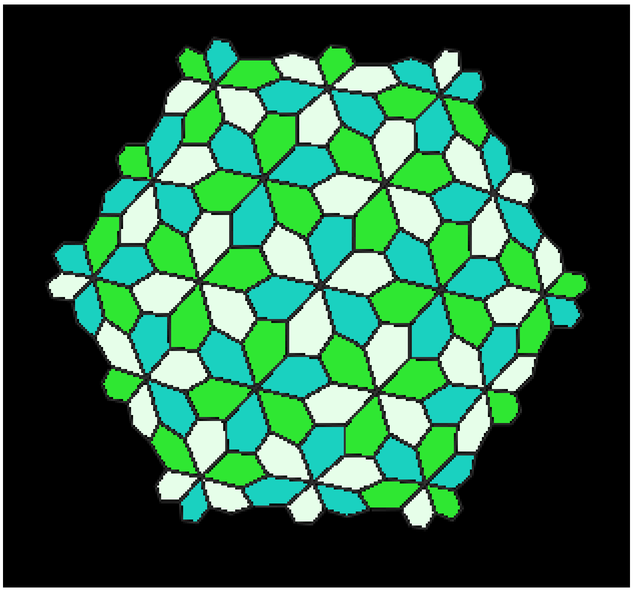
some tiles indicating the tiling

Figure 7.63: *Pentagonal Prismatic Tiling*

vertices: 78, *faces:* 38+1, *edges:* 125, *genus:* 0
characteristic polynomial:
dual's characteristic polynomial:
chromatic number: 3
lattice group (lossy): $Z(2)^{89} \times A(88)$

7.32.2 Pentagonal Floret

This periodic tiling can as well be achieved by starting with a tiling due to Reinhardt.



some tiles indicating the tiling

Figure 7.64: *Pentagonal Floret Tiling*

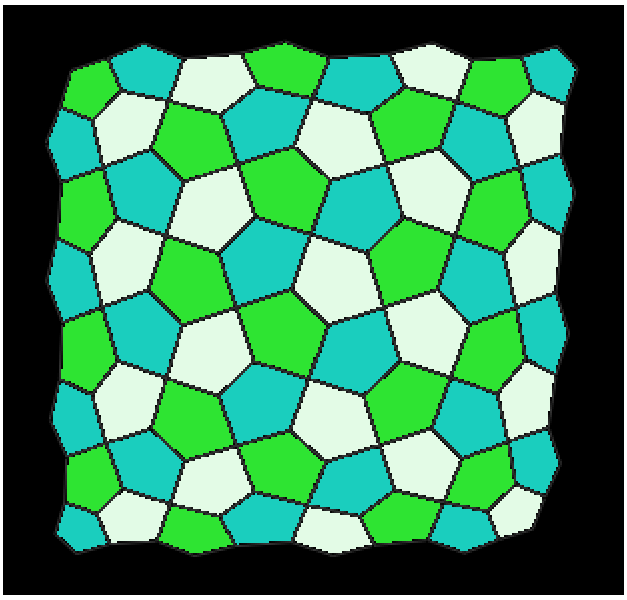
vertices: 217, *faces:* 114+1, *edges:* 330, *genus:* 0
characteristic polynomial:
dual's characteristic polynomial:

chromatic number: 3

lattice group (lossy): $Z(2)^{87} \times Z(3)^{86} \times A(86)$

7.32.3 Pentagonal Cairo

This periodic tiling can be seen as the union of two flattened perpendicular hexagonal tilings. Each hexagon is divided into four pentagons. It actually can as well be achieved by starting with a tiling due to Reinhardt, and another due to Kershner, and it is also basically the same as the Basketweave tiling (if you abstract from unimportant things like lengths and angles).



some tiles indicating the tiling

Figure 7.65: *Pentagonal Cairo Tiling*

vertices: 120, *faces:* 64+1, *edges:* 183, *genus:* 0

characteristic polynomial:

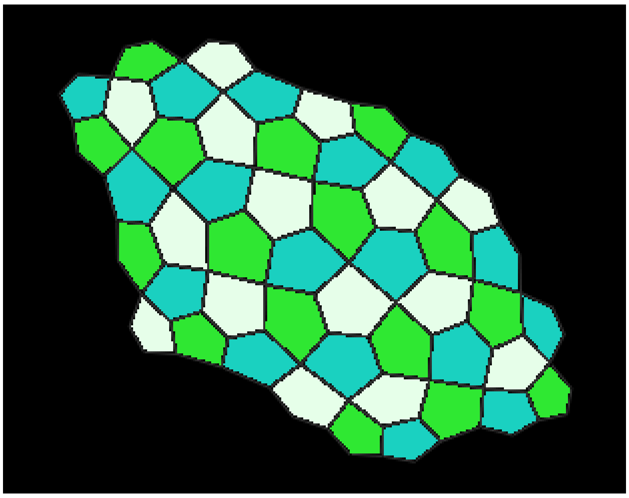
dual's characteristic polynomial:

chromatic number: 3

lattice group (lossy): $Z(2)^{156} \times A(156)$

7.32.4 Pentagonal Rice_I

This periodic tiling is due to M. Rice. It can also be achieved stating from a standard tiling due to Kershner. It seems to be similar to the cairo tiling in that some bigger tile delivers smalles tiles (this time six instead of four).



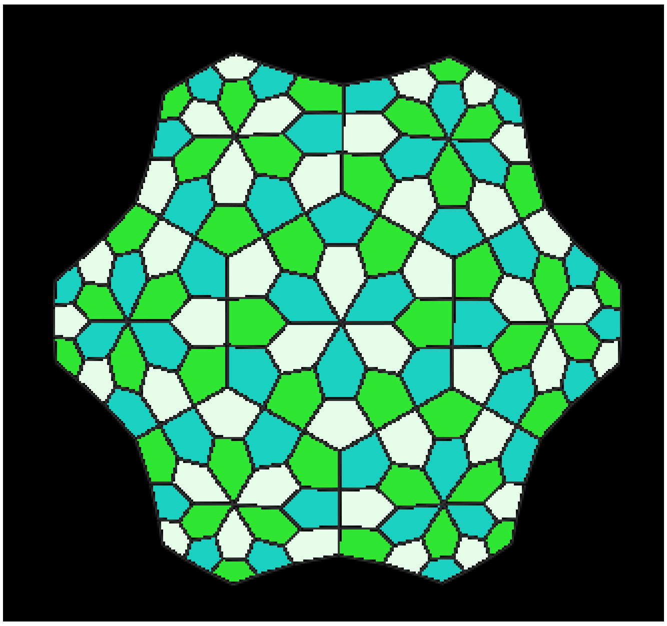
some tiles indicating the tiling

Figure 7.66: *Pentagonal Rice_I Tiling*

vertices: 95, *faces:* 48+1, *edges:* 142, *genus:* 0
characteristic polynomial:
dual's characteristic polynomial:
chromatic number: 3
lattice group:

7.32.5 Pentagonal Rice_II

This periodic tiling stems also from a monohedral tiling due to M. Rice. Here this results in three different sorts of tiles, dependent on where the tiles are located. It is a periodic tiling since it simply decorates a hexagonal lattice. The inner circle of 18 faces is also known as the Hirschhorn Medallion.



some tiles indicating the tiling

Figure 7.67: *Pentagonal Rice_II Tiling*

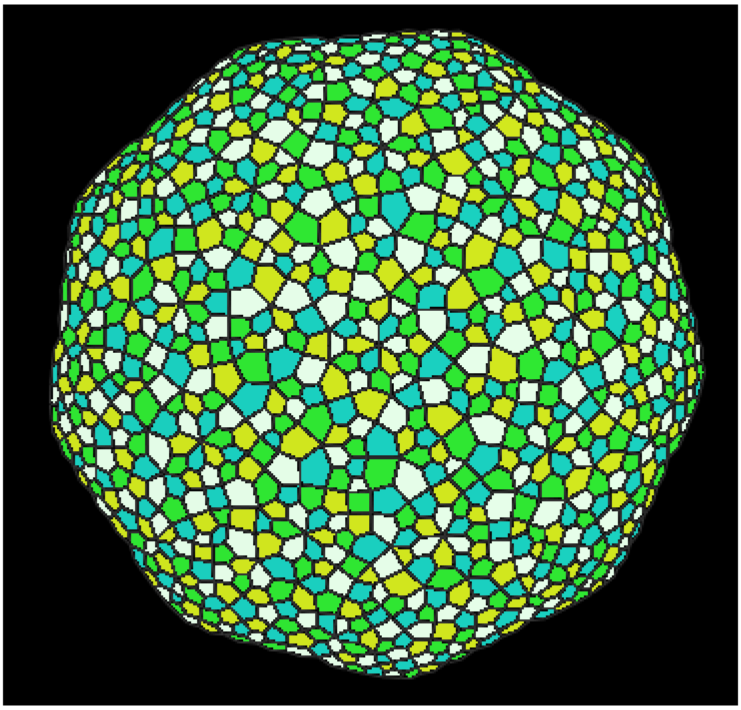
vertices: 217, *faces:* 126+1, *edges:* 352, *genus:* 0
characteristic polynomial:
dual's characteristic polynomial:

chromatic number: 3

lattice group:

7.32.6 Pentagonal Pegg

This aperiodic tiling stems from a duohedral tiling due to Ed Pegg jun. The inner part is the same as in the rice_II example from above. Somehow the flat characteristic is inherent. Here faces connected over multiple edges appear as well.



some tiles indicating the aperiodic tiling

Figure 7.68: *Pentagonal Pegg Tiling*

vertices: 1405, *faces:* 891+1, *edges:* 2295, *genus:* 0

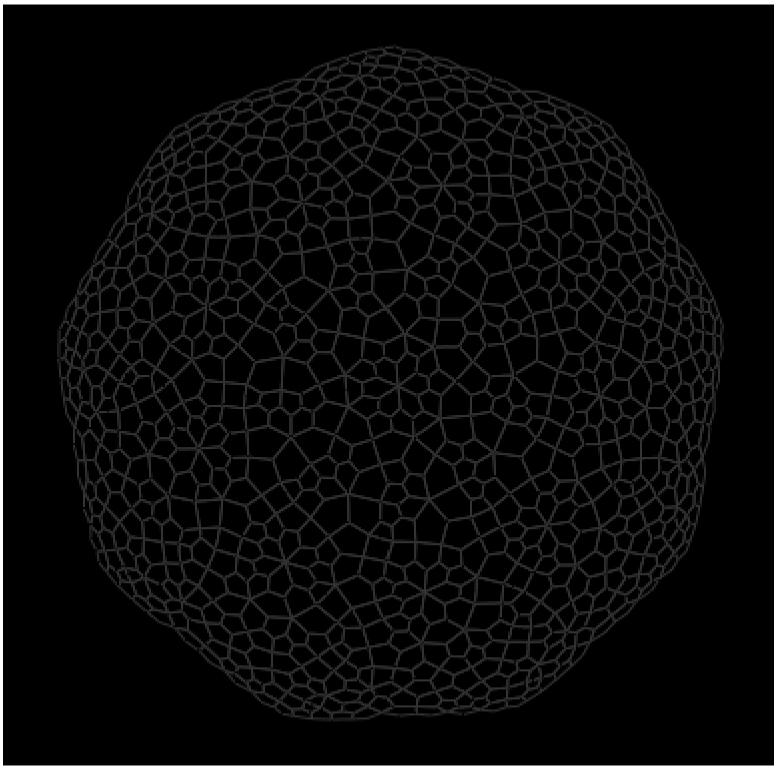
characteristic polynomial:

dual's characteristic polynomial:

chromatic number: 4

lattice group:

Here again the underlying lattice.

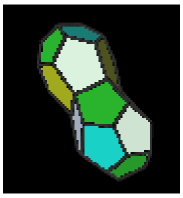


Pegg lattice without irritating colors

Figure 7.69: *Pentagonal Pegg Tiling Lattice*

7.33 DoubleDodecahedron

Two dodecahedra glued together.



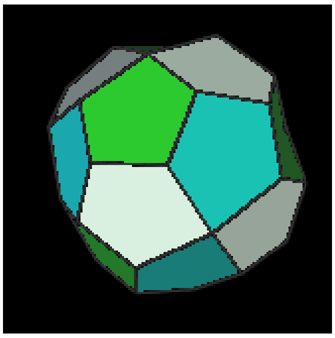
two dodecahedra glued together

Figure 7.70: *DoubleDodecahedron*

vertices: 35, *faces:* 22, *edges:* 55, *genus:* 0
characteristic polynomial:
dual's characteristic polynomial:
chromatic number: 4
lattice group (lossy): $Z(2)^{11} \times Z(5)^{11} \times A(11)$

7.34 Icositetra

Icositetra is Greek and means 24 (the number of faces).



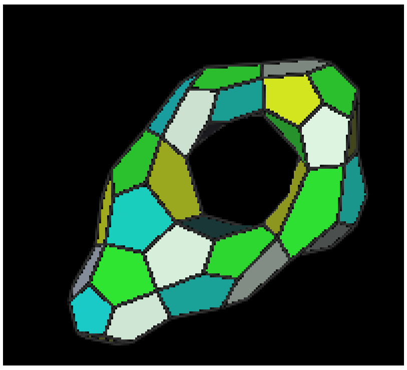
pentagonal icositetrahedron

Figure 7.71: *Icositetra*

vertices: 38, *faces:* 24, *edges:* 60, *genus:* 0
characteristic polynomial:
dual's characteristic polynomial:
chromatic number: 3
lattice group (lossy): $Z(2)^{15} \times Z(3)^{5} \times A(5)$

7.35 Dode Tor 6

A potato-shaped asteroid after the collision with a black hole. Or so.



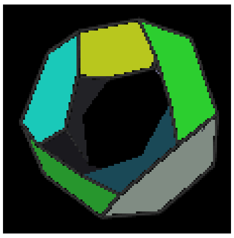
six 'well-tempered' glued dodecahedra

Figure 7.72: *Dode Tor 6*

vertices: 87, *faces:* 58, *edges:* 145, *genus:* 1
characteristic polynomial: (-2836 - 827961x + 13431718x^{2} + ... + 10092x^{83} - 145x^{85} + x^{87})
dual's characteristic polynomial: (-5 + x) (-357491 - 605791 x + ... again quite weird term ... + 5x^{56} + x^{57})
chromatic number: 4
lattice group (lossy): $Z(2) \times A(290)$

7.36 Torus Hexa 3x5

There are toroidal constructions not involving squares.



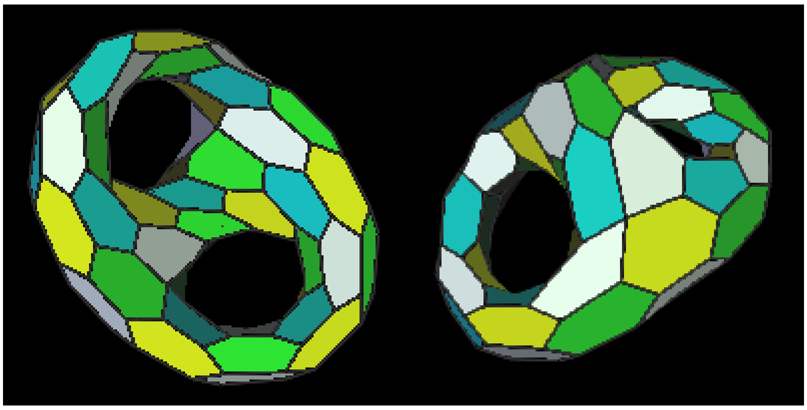
hexagonal torus

Figure 7.73: *Torus Hexa 3x5*

vertices: 30, *faces:* 15, *edges:* 45, *genus:* 1
characteristic polynomial:
dual's characteristic polynomial:
chromatic number: 4
lattice group: ((C15 × C15) : C3) : C2

7.37 Hexa Tor

This graph needs a handful of colors.



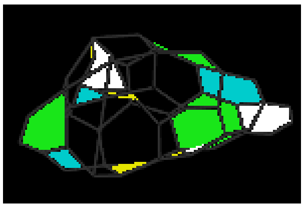
hexagon tesselation with simple lattice group

Figure 7.74: *Hexa Tor*

vertices: 142, *faces:* 72, *edges:* 216, *genus:* 2
characteristic polynomial: (-1 + x) (1 + x) (76312930726164198564 - 6342391150683936840x - ... - 215x^{138} + x^{140})
dual's characteristic polynomial: (-6 + x) (2 + x) (-2243909606400 - 55768207537152x - ... - 188x^{68} + 4x^{69} + x^{70})
chromatic number: 4
lattice group: A(432)

7.38 Hexa UnOrient

We have doubled the original numbers. This example verifies that a closed non-orientable manifold cannot be embedded in $\mathbb{R}^3$ without self-intersection. We could furthermore create another graph with the same (doubled) number of faces, edges and points where we would not enforce that the resulting layout must use the same $\mathbb{R}^3$ locations for both orientations of the original graph, then we have just another standard graph with the same group but this time it's orientable (for a proj_plane graph on 2*10 faces this construction gives just the icosahedron for example).



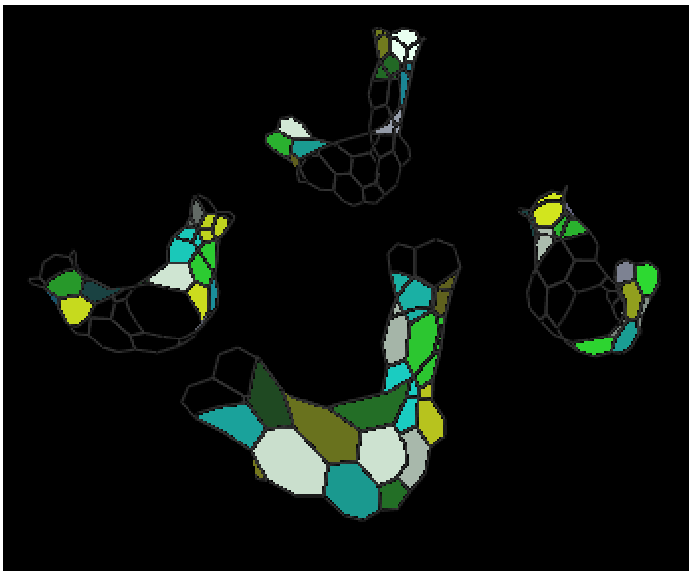
a non-orientable surface with self-intersection

Figure 7.75: *Hexa UnOrient*

vertices: 2*50, *faces:* 2*25, *edges:* 2*74
characteristic polynomial (btw. do we get non-real roots now, if we identify the ids of both orientations?):
dual's characteristic polynomial:
chromatic number: 4 (?)
lattice group (doubled vertices and faces, and dealt with 'connected over multiple edges' issue, lossy version): $Z(2)^{76} \times A(75)$

7.39 Fancy Heptas

This graph is an example where definetely several possibilities exist for 'correct' layouting (the two main parts oriented the same, or (as shown) one inverted).



a (somewhat provisorical) fancy heptagons example in mirror view

Figure 7.76: *Fancy Heptas*

vertices: 90, *faces:* 24+4, *edges:* 116, *genus:* 0
characteristic polynomial: $(-1 + x)^2$ $(1 + x)^2$ $(-1 - 3x + x^2 + x^3)^2$ $(13 - 1155x - ... - 3x^{39} + x^{40})$ $(-9 - 173x + ... + x^{39} + x^{40})$
dual's characteristic polynomial: $(-2 + x)$ $(2 + x)^3$ $(-2 + 69x + ... - 2x^9 + x^{10})$ $(-2 + 47x + ... - 2x^9 + x^{10})$
chromatic number: 4
lattice group (lossy): $Z(2)^{80} \times A(80)$

7.40 ToDo

- ...

Chapter 8

Graphs and Combinatorics

8.1 Adjacency matrix

Using the ExportData utility (with -m option, see Appendix A) and GAP one can compute the expanded form of the characteristic equation, $\det\lambda \mathrm{I} - \mathrm{A} = 0$, where A is the adjacency matrix of a graph. Another interesting thing is the incidence matrix. Call it B, then $BB^T - 2\mathrm{I}$ is the adjacency matrix of the line graph. The Laplacian of a graph is also interesting.

Actually while being first of all primarily interested in the faces instead of the vertices the duals of our graphs are more important to us. Boundaries thereby mean that holes are not regarded as valid but get ignored.

Example: Let's again consider a tetrahedron:

```
gap>A:=[
  [0, 1, 1, 1],
  [1, 0, 1, 1],
  [1, 1, 0, 1],
  [1, 1, 1, 0]
];;
gap>Collected(Factors(CharacteristicPolynomial(A)));
[ [ -3+x_1, 1 ], [ 1+x_1, 3 ] ]
```

See Section 11.5 for a subsequent investigation of those polynomials. For the cartographic version of L2(7) (see Section 7.16) we have

```
gap>A:=[
  [0, 2, 0, 0, 0],
  [2, 1, 2, 1, 1],
  [0, 2, 0, 1, 1],
  [0, 1, 1, 0, 0],
  [0, 1, 1, 0, 0]
]
gap>B:=[
  [2, 1, 0, 0],
  [1, 0, 1, 1],
  [0, 1, 0, 0],
  [0, 1, 0, 0]
]
```

, so you realize the matrices are symmetric and have integral values. Btw. there are also some collapsed adjacency matrices to be discussed.

8.2 Coloring

We focus on the faces and not the vertices. There are lower and upper bounds [1] for the chromatic number χ (the minimal number of colors) dependent on genus, girth and so on. Especially interesting

[1] See the references in Appendix C bibliography division

are graphs with large chromatic number and large girth. Btw. $\chi \geqq$ the clique number (size of the largest complete subgraph).

The famous 4-color theorem states that four colors are always sufficient for a planar graph. There exists a generalization for the maximum number p of colors needed on an orientable surface of genus g :

$$p = \lfloor \frac{7+\sqrt{1+48g}}{2} \rfloor$$

But it remains an NP - complete problem (see Chapter 19) to find out for a given graph if perhaps an even better solution smaller than the theoretically guaranteed bound exists. A greedy coloring-algorithm is not guaranteed to find optimal solutions. Actually it is known that for example the icosahedron can be appropriately colored with only 3 colors, and also for Penrose tilings 3 colors do suffice (see [15]). Therefore some probabilistic methods (see [17]) are preferred. Another possibility might be based on backtracking.

We offer a quite simple 'naive probabilistic coloring procedure' algorithm (one probabilistic round followed by an uncoloring in case of conflicts and finally a greedy fillment) and a more sophisticated approach where in case of conflicts faces placed near the center of the graph are changing colors with a lower probability than more distant faces (the latter is especially more suitable for quasicrystals). Improved methods could for example involve eagier (more than necessary) throwing away of in principle valid color values to establish a smooth, slow but perhaps more substantial construction as well as multiple iterated randomized runs instead of the greedy end.

Take our achieved results about the chromatic numbers with a grain of salt (there are cases where a deterministic approach immediately finds the optimal result whereas our probabilistic algorithm has got problems).

In a next step one could construct the chromatic polynomial containing the number of possible solutions for various numbers of overall colors. And there are connections (see [35] to the so called (Di)Chromatic, Tutte and Rank (edge deletion and contraction) polynomials of a graph. Besides spanning trees, arc-transitivity and graph parameters such as the size of the largest clique or the size of the largest independent set (that is inducing an empty subgraph) etc. could be determined.

The Pólya counting theorem and cycle index stuff establish a connection to group theory (see [52]. Let g be a permutation of some label set Ω (faces, vertices of edges in our case) and $c_k(g)$ the number of k-cycles in the cycle decomposition of g. Besides we put

$z(g) = x_1^{c_1(g)}\ x_2^{c_2(g)}\ ...\ x_n^{c_n(g)}$

where x_1, x_2 ... are indeterminates. Let G be a permutation group, then the polyhedra cycle index is defined as follows:

Equation 8.1 Polyhedra Cycle Index

$$Z(G) = \frac{1}{|G|} \sum_{g \in G} z(g)$$

We are interested in counting the orbits of G on the set of functions from Ω to some set F. Without mentioning any further details of combinatorial enumeration we now simply present the standard Z(G) results for the regular polyhedra rotation groups (acting on the vertices):

graph	Z(G)
tetrahedron	$1/12\ (x_1^{4} + 8x_1x_3 + 3x_2^{2})$
octahedron	$1/24\ (x_1^{6} + 6x_1^{2}x_4 + 3x_1^{2}x_2^{2} + 6x_2^{3} + 8x_3^{2})$
cube	$1/24\ (x_1^{8} + 8x_1^{2}x_3^{2} + 9x_2^{4} + 6x_4^{2})$
icosahedron	$1/60\ (x_1^{12} + 24x_1^{2}x_5^{2} + 15x_2^{6} + 20x_3^{4})$
dodecahedron	$1/60\ (x_1^{20} + 20x_1^{2}x_3^{6} + 15x_2^{10} + 24x_5^{4})$

Table 8.1: *Polyhedra Cycle Index*

In how many different ways (up to rotations) can the vertices of an octahedron be colored with m different colors? Taking each color to be a figure of weight 0, the figure-counting series is simply m, and the number of orbits is

$1/24\ (m^6 + 3m^4 + 12m^3 + 8m^2)$.

Well, of course those aren't all proper colorings, and in general we better work on edge labels for the groups and perhaps restrict the colorings for the edges starting from the same face to have identical value (thereby specifying some face coloring).

See [96] for an application of Gröbner bases and Buchberger's algorithm dealing with a 3-coloring, but beware of the theoretical computational complexity (see as well Chapter 19).

8.3 Combinatorical aspects

Eulerian / Hamiltonian paths and integer sequences in general should be considered.

8.4 ToDo

- various coloring algorithms, dichromatic polynomial
- Eulerian / Hamiltonian paths
- given 2 faces, tell distance for pieces like those of type C (like a Chess knight), and given a start and end direction, construct the shortest (geodesic) path
- how a Chess knight(2,1) on some torus(m,n) sees the world (infile with poly=8 to get the equivalent polyhedron for a piece of type A)
- inverse problem: give matrices or polynomials and recontruct the graph (several graphs in case there is no unique solution)
- verify the Kuratowski theorem: any non-planar graph contains K_5 or $K_{3,3}$
- what about the chain of polynomials constructed by using a graph with boundary, and then taking duals successively, thereby reducing the graph step by step (after 2 steps the inner boundary lines are gone, replaced by their neighbour lines, not necessarily meaning that always faces are removed because they may be shared)
- when making a face invalid: how to achieve the resulting characteristic polynomial from the original?
- optionally take holes as valid faces in polynomials creation
- take the game graph (positions are the vertices with moves as edges) and create characteristic polynomials
- in order to be able to construct groups and polynomials for graphs with faces connected over multiple edges (see the Rubik cube 3x3x3 after 2x2 rotation and hexa_unorient) modify the used methods accordingly (the adjacency matrix entries are the number of connecting edges, so instead of only 0 or 1 now there other integers like 2 are allowed)

Chapter 9

Algebraic, Difference and Geometric Topology

9.1 Euler formula

You possibly already know the Euler formula (Equation 9.1) in three dimensions (v: vertices, e: edges, f: faces, g: genus, 2 - 2g: Euler characteristic):

Equation 9.1 Euler formula

$v - e + f = 2 - 2g$

The genus is a topologic invariant property of the surface. It is defined as the largest number of non-intersecting simple closed curves that can be drawn without separating it. In cases with boundaries holes count as normal faces in fulfilling the Euler formula.

9.2 Homology and Cohomology

It's time to get acquainted with a little bit of homological algebra. See [33] and [30] we look at a sequence of homomorphisms (called boundary operators) of Abelian groups

$$\cdots \to C_{n+1} \underset{\partial_{n+1}}{\to} C_n \underset{\partial_n}{\to} C_{n-1} \to \cdots$$

with $\forall n\colon \mathrm{Im}\,(\partial_{n+1}) \subseteq \mathrm{Ker}\,(\partial_n)$, so the composition of any two consecutive ∂ operators is the constant map to the group identity. Elements of $\mathrm{Ker}\,(\partial_n)$ are called cycles and elements of $\mathrm{Im}\,(\partial_{n+1})$ are the boundaries. $H_n = \frac{\mathrm{Ker}(\partial_n)}{\mathrm{Im}(\partial_{n+1})}$ is the n-th homology group. A chain complex is said to be exact if $\forall n\colon \mathrm{Im}\,(\partial_{n+1}) = \mathrm{Ker}\,(\partial_n)$, so the homology groups measure to what extent the chain complex is not exact.

In case of 2-complexes then $\forall n, n \geq 3\colon H_n = 0$.

For example a torus (one vertex, three edges and two 2-simplices are sufficient for a basic construction) has got (as integral homology) $H_0 = \mathbb{Z}$, $H_1 = \mathbb{Z} \oplus \mathbb{Z}$ and $H_2 = \mathbb{Z}$.

The n-th Betti number (see also Chapter 14) is the rank of the n-th homology group. For a closed, orientable surface of genus g, the Betti numbers are p_0 = 1, p_1 = 2g, and p_2 = 1. The Euler characteristic can be expressed in terms of homology (9.1).

$$2 - 2\mathrm{g} = \sum (-1)^n \mathrm{rank}(H_n(X)) \tag{9.1}$$

Poincaré duality (see [30]) means $\mathrm{H}_i \cong \mathrm{H}^{\mathrm{n-i}}$, De Rham cohomology is defined as closed forms ($\partial\ \omega = 0$) modulo exact ones ($\omega = \partial\ \mathrm{f}$).

9.3 Homotopy

Mapping the homotopy class of each loop at a base point to the homology class of the loop gives a homomorphism from the fundamental group $\pi_1(\mathrm{X})$ (also called first homotopy group; group operation is concatenation, and the inverse is delivered by path reversal) to the homology group $H_1(X)$. X is path-connected so this homomorphism is surjective and its kernel is the commutator subgroup of $\pi_1(X)$, and $H_1(X)$ is therefore isomorphic to the Abelianization of $\pi_1(X)$.

Any finitely presented group can be realized as π_1 of a 2-complex. See [62]) for more information about the fundamental group, the first homology group and the cover of finite 2-dimensional simplicial complex.

Regular coverings correspond to normal subgroups, so every connected G-covering $Y \to X$ has got $G \cong \mathrm{Aut}(Y/X)$ isomorphic to the fundamental group modulo a subgroup (similar to Galois theory, where subgroups correspond to field extensions, smaller subgroups belong to larger extensions). The universal cover is simply-connected and the subgroup is trivial.

The Seifert-van Kampen gives a method for computing the fundamental groups of spaces that can be decomposed into simpler spaces whose fundamental groups are already known.

9.4 Cohomology with Coefficients

$H^n(G, M)$ is called the n-th cohomology group of G (with coefficients in M) involving cocycles and coboundaries.

The group of all functions from the set of components of a complex (these functions are constant on each component) is one interpretation of $H^0(G, M)$.

9.5 Exact Sequences

Mayer-Vietoris sequences (dealing with the union of spaces and their homology) can be viewed as analogs of the Seifert-van Kampen theorem. Besides every short exact sequence

$0 \to A \to B \to C \to 0$

of chain complexes gives rise (using Mayer-Vietoris, Snake lemma) to a long exact sequence of homology groups:

$\cdots \to \quad H_n(A) \to H_n(B) \to H_n(C) \to \quad H_{n-1}(A) \to H_{n-1}(B) \to H_{n-1}(C) \to \quad H_{n-2}(A) \to \cdots$

And finally product spaces can be handled by using Künneth formulas.

9.6 Homology Spheres

Take the unit 3-sphere and form S^3 / 2.A(5), then this is an integral homology 3-sphere and called the Poincaré homology 3-sphere (see [44] for more information). Btw. it's also related to the trefoil knot.

9.7 ToDo

- compute relative homology groups, H_n and so on for all our graphs
- Have a look at the GAP packages HAP for homological algebra (see [66]), homology for (simplicial) homology and simpcomp for working with simplicial complexes.

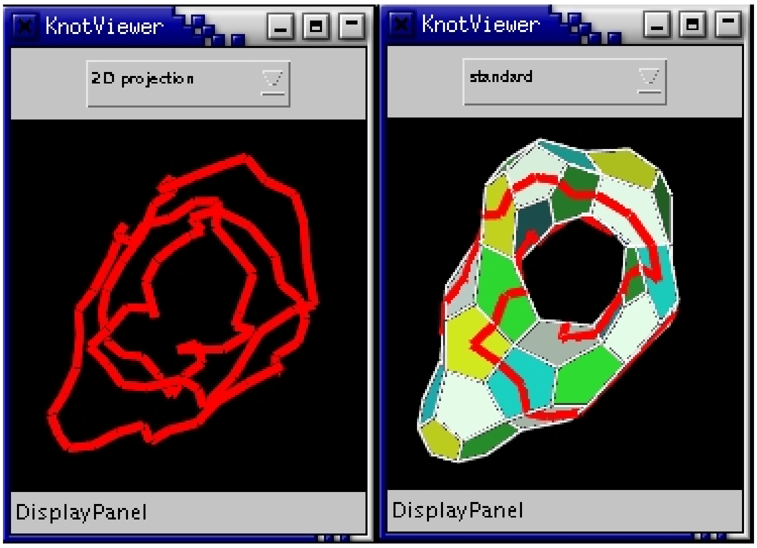
Chapter 10

Knots and Links

10.1 Generalized Torus Knots

Knot theory has to do with our project, for example one can have knots and links embedded on general graphs.

The trefoil knot can be embedded on a torus without any crossings left. Here are some preliminary screenshots (using our KnotViewer utililty) showing a trefoil knot on a 'dode_tor_6' torus and the projection:



KnotViewer displaying a trefoil on a graph of genus 1 and in projection

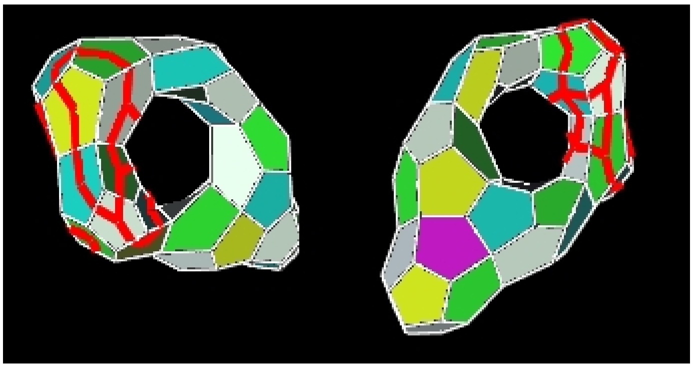
Figure 10.1: *KnotViewer Trefoil*

Unfortunately the three crossings are not identified at one glance, and the overall impression in these pictures is not that overwhelming since whereas topologically all is correct, here the strip goes around the torus in a non-straight artificial manner.

So called (p,q) torus knots and their generalizations are discussed here (see for example [35] and [30]). On a torus the numbers (p,q) are retrieved counting intersections with standard coordinate lines. The link crossing number is then min(p(q-1), q(p-1)).

This is actually using a word in the generators (a,b) to specify how the path should go (see Section 11.2). Since we need also specify if a crossing is over or under we could focus on alternating knots (other rules and levels are possible).

Here (Figure 10.2) is some screenshot with generator "aab".



"aab" knot with 6 crossings, writhe -2 and 3 levels on the "Dode Tor 6" graph

Figure 10.2: *Dode Tor 6 knot*

Exercise (worth 14 points): Prove, refute or declare as undecidable the following conjecture: all knots can be embedded as an alternating knot on a discrete 2-manifold with suitable group and genus.

Maybe the conjecture should even postulate embeddings on graphs without any crossings left (like the trefoil on a torus). Or the sort of crossings could be determined from the Schur cover sheets.

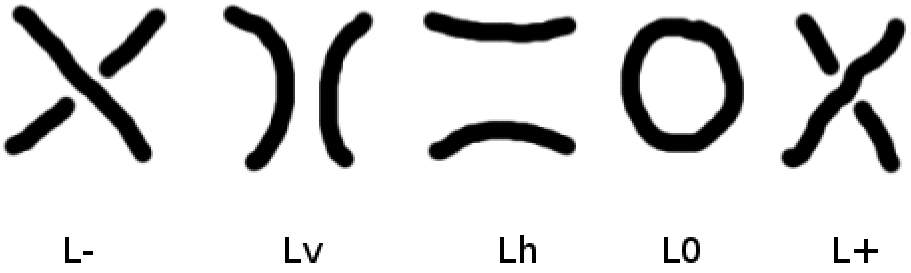
It is known (see for example [40]) that every knot (and link) is n-embeddable (on surface of genus n), every alternating knot is toroidal alternating, and every fast-alternating knot is toriodal alternating as well.

10.2 Invariant Polynomials

Loops without any crossing left deserve the same role the circle has got in the standard plane, so we might compute polynomials relative to our graphs. That is to say the Reidemeister movements are (discretely) done on our manifolds and not in the projection. And (we hardly dare to guess, but) of course those movings ropes around (without cutting) form some group to be investigated.

Using again the ExportData utility (-kd and -kp) you get the path data and (by the help of our preferred computer algebra system afterwards) the generalized Jones polynomial using the writhe and some skein tree.

We use the following notation:



knot diagram labelling

A crossing L+ has got a signature +1 (and L- gets -1). The writhe `w` is the sum of all signatures.

Let `L` denote the general link, and `<L>` the polynomial. The rules below can be applied (making the polynomial X (but not L) invariant under the three Reidemeister movements):

```
Rule 1: < L0 > = 1

Rule 2: < L- > = A < Lh > + A^-1 < Lv >
        < L+ > = A < Lv > + A^-1 < Lh >

Rule 3: < L L0 > = (-A^-2 - A^2) < L >
```

The Jones (a special case of the more general Kauffman) polynomial J(L) is retrieved by adding the factor

$X = (-A^3)^{-w} <L>$

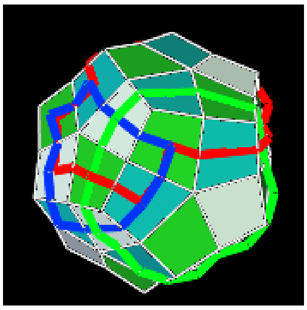
,and finally a substitution is done:

$t := A^{-4}$

It is an open problem whether J(K) = 1 implies that K is unknotted for some knot K. The corresponding question for links has already been settled, there are many 2- and 3-component links whose Jones polynomial is equals to that of the corresponding unlink. On the other hand, there are many pairs of knots (for example the Kinoshita-Terasaka knot and the Conway knot) having the same Jones polynomial but being topologically distinct. So the Jones polynomial is not a complete invariant.

In order to do those polynomials in our graphs (instead of on the standard plane or cube, that is to say on s.th. with genus 0) we modified the rules a little bit. We apparently have to specify some more atomic results in addition to the rule one, for a loop on a torus without crossings may represent a trefoil and not the trivial unknot. In rule 3 (replacing the 1 there) a multiplication with the polynomial of the loop should take place. And the overall writhe should on one hand consist of the things visible on the very graph itself as well as the parts belonging to the loops. We lose crossings in between different loops (in skein tree and writhe) (and this may actually be the gain of the task and the reason to do the procedure on a generalized graph).

On genus 0 graphs we get the standard results of course. See for example the Borromean rings (6_2^3 in Rolfsen notation) with writhe 0 (yes, it looks a little bit strange, we admit that) with the polynomial V(t) (lowest exponent and coefficients): {-3}[-1,3,-2,4,-2,3,-1]

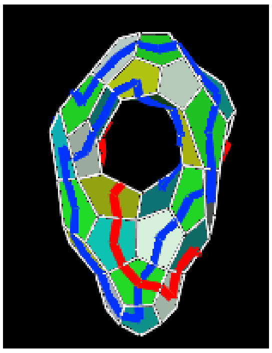


Borromean rings on a graph of genus 0

Figure 10.3: *Borromean rings*

The Borromean rings example is the simplest possible Brunnian link. A Brunnian link is a nontrivial link that becomes trivial (use the polynomial to detect that) if any component is removed. For every number three or above, there are an infinite number of Brunnian links containing that number of unknots. General n-Borromean links are defined as n-component nontrivial links such that any two components form a trivial link, but with at least one nontrivial sublink (thereby being distinguished from the Brunnian links in which every sublink is trivial).

Let us take an easy example now (a self-brewed link with one crossing, the skein tree stops at a trefoil and a (1,1) unknot) to speak about our more general polynomials. Here is a screenshot:



a link with one crossing on the graph "Dode Tor 6" of genus 1

Figure 10.4: *Dode Tor 6 Link*

We paid attention to the fact that paths without any remaining crossings on a torus can in fact be different from the trivial unknot, so we left those loop polynomials and the writhes as variables to be set manually (inserting $-A^{-5} - A^{3} + A^{7}$ for the trefoil L_0 and -3 as its writhe w_0 contribution). It seems that a (1,1) torus knot (which is actually standing alone the unknot as well) should get s.th. like A (and not simply 1) and writhe 0. Then we get as polynomial V(t):

{5}[1,-1,1,-1]

and

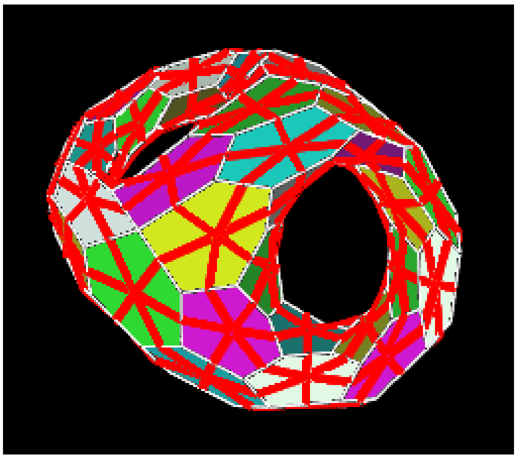
{-9/2}[-1,0,0,1]

in the other direction respectively replacing all A with its inverse.

Otherwise sometimes some unclear things are going on in the procedure, leading to broken exponents even for initial knots (for links this may often happen) and losing of coefficients in the final crude substitution.

Well, this will perhaps be clarified in the future.

Exercise (worth 5 points): Compute the Jones polynomial of the colorful flower power knot in Figure 10.5.



"ab" knot cover with 578 crossings (?!), writhe 67 (?!), 6 levels on "Hexa Tor" graph of genus 2

Figure 10.5: *Hexa Tor Knot*

Instead of using constant word ("ab" or "aab") one may construct paths using a constant acceleration or other things. The methods could also be applied to links in case of multiple pieces walking around. Besides there are other polynomial invariants (HOMFLYPT etc.) to be dealt with.

Vertex operator algebras might give further insights?!

10.3 ToDo

- in KnotViewer indicate the orientation (direction)
- perhaps interactive Reidemeister moving lines around
- discrete Reidemeister movements build a finite group, figure out a construction for this
- implement the finishing parts for the knot treatment on graphs with boundary
- dichromatic polynomial and connections to graph coloring
- symmetry group of a link (see references Z(2) × A(5) and others)
- what happens when we take those polynomials seriously and let the variable having values in some field
- the running time and space requirements for the used Jones polynomial computing algorithm are in what class? O(n sqrt(n)) or s.th. like that (n denoting the number of crossings)?

- formula for Jones polynomial of (p,q) torus knot can be retrieved using algebras; well, so there is hope that our computational approach can serve as a starting point for general guesses and theorems (avoiding the skein tree at all and just using general formula)
- classify Chess piece movements (multiple pieces trying to kill the kings, resulting in a periodical behavior and crossed paths) in the context of links
- use MathML and special fonts designed for knot diagrams

Chapter 11

Group Theory

11.1 General Theory

Consider the group (lattice group, automorphism group) generated by the following two actions on directed edges:

- face rotation
- edge flip

By using the export utility (again see Appendix A) we can get these generators. A third operation, the rotation of directed edges (those starting from there) around the vertices (counter-clockwise), can be achieved as $dual\_rot * rot = \text{flip}^{-1}$

In the case of holes we use the 'double' (graph, glueShapes along inner boundary lines, and mirrored graph, see Section 11.2 for an explanation). One could also take holes simply as valid faces. Even though topologically our graphs only differ by genus, geometrically some vertices are different from others, their valences vary. How about the relationship of stellation, gluing or turning faces to holes and the resulting groups?

Example: Let's consider a tetrahedron:

```
gap>G:=Group(
  (1, 2, 3)(4, 5, 6)(7, 8, 9)(10, 11, 12),
  (1, 11)(2, 6)(3, 9)(4, 10)(5, 7)(8, 12)
);;
gap>Collected(Factors(Size(G)));
[ [ 2, 2 ], [ 3, 1 ] ]
gap>IsNaturalAlternatingGroup(G);
false
gap>H:=AlternatingGroup(4);;
gap>l:=NormalSubgroups(H);
[ Alt( [1..4] ), Group([ (1,3)(2,4), (1,2)(3,4) ]), Group(()) ]
gap>OneCocycles(H,l[2]);
... isSplitExtension:=true;
gap>cmpl:=Complementclasses(H,l[2]);
[ Group([ (2,4,3) ]) ]
gap>Size(Intersection(cmpl[1],l[2]));
1
gap>ClosureGroup(cmpl[1],l[2]);;
gap>IsomorphismGroups(H,G);
[ (1,2,3),(2,3,4) ]
  -> [ (1,9,12)(2,11,4)(3,6,7)(5,10,8), (1,2,3)(4,5,6)(7,8,9)/10,11,12) ]
gap>DisplayCompositionSeries(G);
G (3 gens, size 12)
 | Z(3)
S (2 gens, size 4)
 | Z(2)
S (1 gens, size 2)
 | Z(2)
1 (0 gens, size 1)
```

A(4) is a Frobenius group (that's a transitive permutation group on a finite set, such that no non-trivial element fixes more than one point and some non-trivial element fixes a point).

A subgroup is normal if and only if it is invariant under all inner automorphisms.

Please be aware that instead of the exact group in the following we often only present some sorted simple factors in a composition series. For example actually A(4) is (Z(2) × Z(2)) : Z(3) (direct and semi-direct product) whereby Z(3) is *not* a normal subgroup, so just merging the factors into a final big pile is a quite lossy presentation and should be improved.

It seems that the following ATLAS notation should be used: A × B for a direct product; A.B or A B for a group with normal subgroup isomorphic to A, for which the corresponding quotient group is isomorphic to B; A : B for the case of A.B which is a split extension (or a semi-direct product in other words); and $A \cdot B$ for the case when A.B is not a split extension.

A canonical naming is available by GAP (there A.B already means non-split):

```
gap>StructureDescription(G);
'A4'
```

Whereas non-isomorphic groups can get the same description, this is anyway a substantial improvement.

Some cyclic and alternating groups seem to be the standard results. In fact as (the number of labels) $n \to \infty$ the result will almost always be S(n) or A(n). The alternating groups are finite analogs of the families of simple Lie groups (aka 'groups of Lie type'). Anyway there are numerous isomorphisms like A(4) ≅ L2(3) , A(5) ≅ L2(4) ≅ L2(5) , A(6) ≅ L2(9) , and A(8) ≅ L4(2) (thereby for example L2(3) and PSL(2,3) are different notations and denote the same group).

A standard procedure to contruct a minimal faithful (transitive) permutation representation of a group G is to look for the largest maximal subgroup S. Then the index of this group #G/#S gives the number of the points (labels).

Let's create the regular (2,4,5) representation of S(5) ≅ A(5) : 2 (on 120 labels of genus 4):

```
gap>S5:=SymmetricGroup(5);;
gap>hom:=RegularActionHomomorphism(S5);
<action epimorphism>
gap>S5_reg:=Image(hom,S5);
```

The same works for the simple group L2(7) of order 168 and genus 3 using the (2,3,7) triple. Both groups obey the Riemann-Hurwitz identity (Equation 11.1), relating the group's size, the symmetric genus, and the valences 2, p and q (around edges, faces and vertices respectively):

Equation 11.1 Riemann-Hurwitz identity

$$|G| = 2g - 1\left(1 - \frac{1}{2} - \frac{1}{p} - \frac{1}{q}\right)^{-1}$$

Moreover the group achieves the maximal possible order 84(g-1) and so is a Hurwitz group. L2(7) is the automorphism group of Klein's quartic on a set of 24 regular heptagons (or alternatively with a set of 56 equilateral triangles), and it's as well the group of symmetries of the Fano plane.

From the sporadic simple groups, the following twelve possess (2,3,7) generators and are as well Hurwitz groups: J1, J2, He, Ru, Co3, Fi22, HN, Ly, Th, J4, Fi'24 and M. The dessin is just the Cayley graph.

For the sporadic Mathieu group M(24) obeying $\text{rot}^3 = \text{flip}^2 = \text{dual\_rot}^{21}$ a dessin can be constructed with the rot on 6 triangles and the dual_rot having one cycle of length 21 once around the boundary, and one cycle of length 3 (luckily dividing 21) around the single inner vertex.

See [31] let C_k denote the conjugacy classes of the group G and define a number as

$N := \#\{ (c_1, \ldots, c_k) \text{ in } C_1 \times \ldots \times C_k \mid c_1 \ldots c_k = 1 \}$

, then using the complex irreducible characters χ of G we get the generalized Frobenius formula (Equation 11.2) for ramified coverings of a Riemannian surface (without torsion) with genus g.

Equation 11.2 Frobenius formula

$$N = |G|^{2g-1}|C_1|\ldots|C_k|\sum\frac{\chi(c_1)\cdot\ldots\cdot\chi(c_k)}{\chi(1)^{k+2g-2}}$$

For example take the simple group M(24) and the class vector (2A, 3A, 21A) with k=3 (flip,rot,dual_rot), look at the character table and realize g = 0. And btw. all this is also related to the so called normalized structure constant of a class vector.

We can determine the group of automorphisms of a group G (icosahedron G:=A(5) example):

```
gap>AG:=AutomorphismGroup(G);
gap>DisplayCompositionSeries(AG);
G (3 gens, size 120)
 | Z(2)
S (2 gens, size 60)
 | A(5)
1 (0 gens, size 1)
gap>Size(InnerAutomorphismsAutomorphismGroup(AG));
60
```

Here AG has got 7 irreducible characters, and A(5) is the normal subgroup, Z(2) is not.

Let's see what GRAPE gives as the edge-set preserving automorphism group (here A is the icosahedron adjacency matrix):

```
gap>LoadPackage("grape");;
gap>P := Graph( Group(()), [1..20], OnPoints,
      function(x,y) return A[x][y]=1; end, true);;
gap>G:=AutGroupGraph(P);;
gap>DisplayCompositionSeries(G);
G (3 gens, size 120)
 | A(5) ~ A(1,4) = L(2,4) ~ B(1,4) = O(3,4) ~ C(1,4) = S(2,4) ~ 2A(1,4) = U(2,4)\
 ~ A(1,5) = L(2,5) ~ B(1,5) = O(3,5) ~ C(1,5) = S(2,5) ~ 2A(1,5) = U(2,5)
S (1 gens, size 2)
 | Z(2)
1 (0 gens, size 1)
```

This is the direct product Z(2) × A(5) (both Z(2) and A(5) are normal subgroups, G has 10 irreducible characters).

Yet another group is returned when we ask for all matrices commutating with B, the adjacency matrix of A's dual (the dodecahedron):

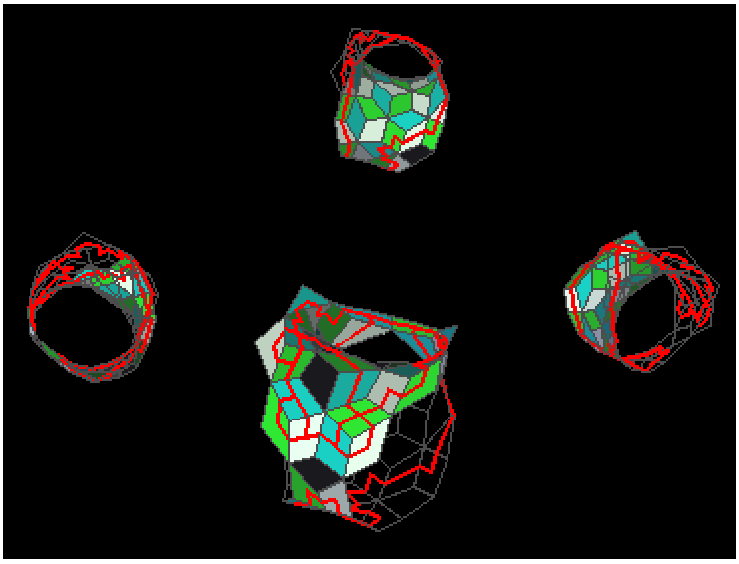
```
gap>G := GL(12,2);
SL(12,2)
gap>ConvertToMatrixRep(B,2);;
gap>C := Centralizer(G,B);;
gap>DisplayCompositionSeries(C);
(A(11,2) = L(12,2)
```

Actually (even more meaningful and important) one might take the rot/flip group from above and represent each element by a n × n permutation matrix with n the number of used labels (the degree), the entries (α , β) being 1 if the group element changes α to β , and 0 otherwise. Such a map is a homomorphism from the group into GL(n,F) for any field F. The centraliser algebra is the set of all n × n matrices which commute with all arising matrices of the group (see [52]). The algebra's dimension equals the rank of the permutation group.

11.2 Holonomy Groups

A Chess move doesn't permute the source and target place. It's more like a vector from the start to the end point. The location of a Chess piece could then be interpreted as a vector starting from an arbitrary origin accompanied by a moved-along local coordinate system (frame).

Let's look at a graph with plenty of holes (Figure 11.1).



three inner boundary lines in mirror view presentation

Figure 11.1: *Holonomy*

The treatment must fulfill the following criteria:

- we want a group, so an inverse must always exist, that's why we can't simply fix things at a boundary (we want a regular parametrization)
- the group definition forbids irreversible behavior: no forgetting about initial values without periodicity
- the result can be mapped (by an epimorphism) to ordinary Chess piece movement (with space-time metric) and some intuitively understood physical 'reflection' at a boundary (with incidence angle and so on)
- aesthetical results (that is to say symmetry)

It is said that (2-)manifolds are looking locally like vector spaces. The idea now is to construct s.th. globally closed under addition, with associative law, identity and inverses, but non-Abelian in the general case. It's the group of parallel transport (holonomy group).

The moves are put into equivalence classes (reflexive, symmetric, transitive), and then we only deal with representatives for each class. One vector stands for many concrete moves later on. The rule is to move the second summand along some coordinate lines (the 'vector's direction or somehow perpendicular), until its start meets the line constructed by first summands' orientation and then a movement along this line upto the first summands end point preserving the relative orientation. We identify 'vector's in case they can be moved onto one another.

Inner boundary lines are good candidates for coordinate lines representing a somewhat appropriate system (btw. pay attention to the fact that they have an orientation, depending on their inner or outer placement). Coordinate lines may locally coincide (degenerate situation).

For the moment let's work with only two generators (poly*2 many could look more natural but are not needed). As an example an m × n torus get's relations $a^n = b^m = [a,b] = 1$ and a composition like Z(m) × Z(n). A square board with boundary, let's say 5 × 6 Chess board, does *not* look the same as some torus board without boundary. Instead involving four 2-gons (can be realized with non-straight edges) we get s.th. of genus 0.

In general boundaries make vectores bending backwards representing reflection in case the approaching vectors seemingly want to enter an hole area. The multiple reflections taking place in case of several holes (possibly with partly coinciding lines) are (for now) supposed to be commutative, and therefore we simply take the graph and it's mirrored instance (experts in manifold theory might name it the 'double'), and then glue together all corresponding inner lines.

Let's take a cube now and see what we can do with it. Define a := flip $\text{rot}^{(\text{poly}/2)}$ (means first flip, then rots afterwards) and b := rot a rot^{-1}. The existence of non-trivial boundaries implies shapes with valences other than poly. By using the dynamic arbitrary ’valence/2’ definition of what is considered straight for the coordinate lines we achieve a nice walking around boundaries.

```
gap>f := FreeGroup("a", "b");;
gap>g := f / [ f.1^4, f.2^4, (f.1*f.2)^3, (f.1^2*f.2^2)^2,
               (f.1*f.2^2)^2, (f.2*f.1^2)^2 ];;
gap>Size(g);
24
gap>IsAbelian(g);
false
gap>a:=g.1;;b:=g.2;;
gap>a*b = b*a;
false
gap>FreeGroupOfFpGroup(g)=f;
true
gap>FreeGeneratorsOfFpGroup(g);
[ a, b ]
gap>hom := IsomorphismSimplifiedFpGroup(g);
[ a, b ] -> [ a, b ]
gap>s := Range(hom);
<fp group on the generators [ a, b ]>
gap>RelatorsOfFpGroup(s);
[ b^4, a^4, a*b^2*a*b^-2, b*a^2*b*a^-2, a^-1*b*a^-1*b*a^-1*b ]
gap>h := IsomorphismPermGroup(g);
[ a, b ] -> [ (2,4,3,5), (1,2,6,3) ]
gap>h2 := IsomorphismPermGroup(s);
[ a, b ] -> [ (1,2,4,3), (2,5,3,6) ]
gap>d := Image(h2, s);
Group([ (1,2,4,3), (2,5,3,6) ])
gap>IsSymmetricGroup(d);
true
gap>s4 := SymmetricGroup(4);;
gap>IsomorphismGroups(d,s4);
[ (1,2,4,3), (2,5,3,6) ] -> [ (1,4,2,3), (1,4,3,2) ]
gap>IsomorphismGroups(s,s4);
[ a, b ] -> [ (1,2,4,3), (1,4,2,3) ]
gap>StructureDescription(d);
’S4’
```

In case of a tetrahedron, a cube/octahedron and an icosahedron/dodecahedron we get, as expected, the standard groups A(4), S(4) and A(5).

The choice of coordinate lines (origin, distIE) must not have any effect on the resulting final group. Of course, same as in linear algebra there are degenerate coordinate systems where not the full group is achieved, for example the icosahedron gets A(5) with distIE_a=2, distIE_b=1, rel_a_b=2 and the same with distIE_a=2, distIE_b=2, rel_a_b=1], but only Z(5) with distIE_a=2, distIE_b=1, rel_a_b=1.

Later on we won’t be especially interested in the actual groups, but only on the local structure. This is where we’ll start to create a Hamiltonian, since things like curvature, [a,b] etc. are important, but if *a*ˆ4 or *a*ˆ7 give the identity targets mostly the global aspects (let’s say the size of the board), this will influence the group, but it is not important to us, so tak it as an indication that the groups are quite important but not the final goal.

Besides recall into your mind that for example on a cube in doing [a,b]=$aba^{-1}b^{-1}$ there are 12 steps and one full counter-clockwise spin, on the standard plane no spin occurs, and on the quad_star (start with a ’vector’ looking against a wall in the middle) you have again 12 steps, but a clockwise spin this time. It seems this is actually called torsion and will turn out to be quite decisive later on in Section 11.3 :-)

It’s a field of study in its own right to investigate what relations (the non-defining additional information are called laws) will be sufficient to present a well-defined finite group in this context. But one should ask oneself if an encoding by generators/relations and subsequent reconstructing by the computer algebra system is a somewhat complicated and indirect procedure.

Actually there is no reason why we shouldn’t dump out a permutation representation directly. Here (Table 11.1) are some resulting generators "a", "b" and groups (often only the lossy version is given) for various graphs.

A cube with one hole gives just the same group as the normal cube. The doublecube has been treated

graph	a	b	group
dode 0,1,5,7,9			A(16)
oct_stl_I			((C4 × C4) : C3) : C2
tetra_reticul			((C6 × C6) : C3) : C2
tetra_reticul_big			((C8 × C8) : C3) : C2
tetra_reticul_tall			((C10 × C10) : C3) : C2
oct_stl_II			$Z(2)^{15} \times Z(3)^{4}$
ico_stl_I			((A5 × A5 × A5) : C3) : C2
ico_stl_II			$Z(2)^{2} \times A(5)^{4}$
doubletetra			(((C3 × ((C3 × C3) : C2)) : C2) : C3) : C2
cube_triang_II			A(36)
M_24	(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21) (22,24,23)	(1,22,17,16,23,20,19,24,14,3,2,13,10,7,6,9,8,5,12,11,4) (15,21,18)	M(24)
doublecube			$Z(2)^{10} \times A(5)$
dblcube 10			$Z(2)^{13} \times A(7)$
dblcube 8			$Z(2)^{25} \times A(26)$
dblcube 6,8	(1,2,3,4) (5,6,7,8,9,10,11,12,13,14,15,16) (17,18,19,20,21,22) (27,26,23,29) (24,32,31,30,25,28)	(5,17,9,13) (6,23,24,1,25,26,12,3,18,27,22,2) (7,11,28,15) (8,21,20,19,10,29) (14,30,31,32,16,4)	$Z(2)^{16} \times A(16)$
dblcube 3,8			$Z(2)^{13} \times A(7)$
dblcube 0,6,8	(1,2,3,4,5,6) (7,8,9,10,11,12) (14,13,15,16)	(7,6,5,8) (9,13,14,12,11,10) (15,4,3,2,1,16)	$Z(2)^{12} \times Z(3)^{2}$
triplecube			$Z(2)^{13} \times A(7)$
rhombic_dodeca			((A4 × A4) : C2) : C2
zonotope_5			$Z(2) \times Z(5)^{8} \times A(8)$
torus_3x4			C12
torus_3x4 1,4,5,6,9			$Z(2)^{10} \times A(10)$
2x3_rectangle			(((C2 × D8) : C2) :C3) : C2
quad_star			$Z(2)^{19} \times Z(3)^{9} \times A(9)$
rubik_cube_2			$Z(2)^{11} \times Z(3)^{4}$
rubik_cube_3			$Z(2)^{19} \times Z(3)^{11}$
rubik_cube_3_1			$Z(2) \times Z(3)^{70} \times A(36)^{2}$
rubik_cube_3_2			$Z(2) \times A(108)^{2}$
costa			$Z(2)^{30}$
quasi_0			C5 × D10 × D10
quasi_1			$Z(2)^{15} \times Z(5)^{15} \times A(15)$
quasi_2			$Z(2)^{53} \times Z(5)^{53} \times A(53)$
quasi_hepta_0			C7 × D14 × D14
dode_tor_6			Z(2) × A(290)
torus_hexa_3			C15
hexa_tor			A(432)
fancy hepta			$Z(2)^{80} \times A(80)$

Table 11.1: *Holonomy Groups*

with one and two outer holes, one inner hole or various possibilities how two inner holes can be placed (neighboured, touching, opposing or asymmetric). With two connected inner holes this gives again just the same group as the untouched doublecube. An outer hole enables a due group treatment with standard Cartesian (canonical) coordinates (only memory limits us), but one inner hole makes severe problems, so there our choice of coordinate lines (as shown above in the nice picture) are probably really necessary.

Pay due attention to the fact that there is no longer a 1:1 correspondence between the labels used in the permutation representation and the group elements, the number of the latter just explodes.

It seems that (for example in the case of the doublecube with holes at faces with ids 6 and 8) there occur singularities (a change from 2-dim to 1-dim)?! We cured it with a regular parametrization involving glueShapes of valence 5, all is quite fishy but works.

The flip/rot groups are not the same as the holonomy groups, on a torus with squares (quadrilateral faces) the latter is not transitive regarding the oriented edges. As you can see (in doublecube, triplecube, ico_stl_II, rubik_cube_3_1 for example), it happens that the 'vector' groups and the rot/flip groups differ by a factor of Z(2), but for the dodecahedron and quad_star this factor is not needed, because there the rotation (linear operation) can be globally expressed as a series of translations.

Let's look at the dodecahedron and the generators for the holonomy group (Example 11.1).

Example 11.1 Dodecahedron Holonomy

```
gap>G := Group(
  (25,29,36,43,6)(1,18,32,47,48)(2,37,40,49,17)(3,51,41,31,35)(54,30,16,22,38)
  (7,26,34,15,23)(10,14,24,39,50)(11,21,45,8,27)(60,13,19,53,57)(20,4,59,42,12)
  (9,28,33,58,5)(52,44,55,56,46),
  (1,2,3,4,5)(18,19,20,21,14)(32,33,11,34,29)(54,40,47,50,26)(6,38,51,57,48)
  (7,55,41,49,43)(10,25,37,53,58)(60,17,22,44,59)(15,24,28,12,46)(9,13,35,52,45)
  (30,36,39,27,56)(42,31,16,23,8),
  (1,6,7,8,9)(18,2,22,23,24)(32,19,35,16,36)(25,38,44,45,14)(3,52,15,29,37)
  (41,57,58,11,56)(10,26,55,59,5)(60,48,50,27,42)(47,33,12,31,49)(20,46,30,40,53)
  (13,17,43,39,28)(54,51,4,21,34),
  (1,10,11,12,13)(18,25,26,27,28)(32,37,38,7,39)(2,6,50,33,19)(3,22,43,47,53)
  (54,55,8,24,29)(60,5,21,46,31)(14,34,56,42,9)(15,30,41,59,45)(20,35,17,48,58)
  (51,44,23,36,40)(4,52,16,49,57),
  (1,14,15,16,17)(18,29,30,31,13)(32,40,41,42,28)(25,34,46,35,2)(3,38,26,11,20)
  (54,56,12,19,37)(6,10,21,52,22)(7,50,58,4,44)(47,57,59,8,39)(51,55,27,33,53)
  (48,5,45,23,43)(60,9,24,36,49)
);;
```

We try to express a rotation on a dodecahedron as a word in those generators.

```
gap>perms:=[G.1, G.2];;
gap>puzzle:=Group(perms);;
gap>Size(puzzle);
60
gap>F:=FreeGroup("a", "b");;gens:=GeneratorsOfGroup(F);;
gap>hom:=GroupHomomorphismByImages(F, puzzle, gens, perms);;
gap>word:=PreImagesRepresentative(hom, G.3);
a^-1*b^-4
```

Check that d = $b^{-1}c^{-4}$ as well, so we do know by now how to express a rotation (btw. we could as well think of symmetric and antisymmetric combinations). In Table 11.2 we expressed c and the local rot as words over (a,b) for some graphs.

We experimented with up to poly generators (for an A-type piece), but (it might be astonishing to the unprepared reader) it turned out that (at least in all examples we investigated) always two generators are enough. A quick rigorous prove is postponed.

A Chess' knight-like piece gets not a mere subgroup but takes other generators (let's start with poly*2 many and then see if again two is all we need). In this way the correct reflection at a boundary can be achieved.

A piece able to walk more than one step (during a single move) will need more generators. Besides "a" and "b" (here the rotation has taken effect as a similarity transformation) we have another generator

graph	c	rot
tetrahedron	$b^{-1}a^{-1}$	$a^{-1}b^{-1}$
cube	a^{-1}	aba^{-1}
dodecahedron	$a^{-1}b^{-4}$	aba^{-2}
doubletetra	$a^{-3}*b^{-1}*a^{-1}$	$a^{-1}*b^{-1}$
doublecube	a^{-1}	fail
ico_stl_II	a^{-1}	fail
rubik_cube_3_0	a^{-1}	fail
rubbik_cube_3_1	(out of memory)	fail
rubbik_cube_3_5	(out of memory)	
quad_star	a^{-1}	$a^{-1}*b*a^{-1}*b^{-1}*a* \dots *a^{3}*b*a^{-1}$
torus_hexa_3x5	$b*a^{-1}$	
hexa_tor	(out of memory)	

Table 11.2: *c and rot as words*

resulting from an energy boost operation. Labels of energy 1 are mapped to those with energy 2 and so on.

It seems that the moves of a piece of type D are not invertible, so we don't have a group element here. The same thought applies to capturing moves (see also the talking in Chapter 17 where general game constellations other than drawn ones are discussed).

Let V denote the set of vectors, W the set of faces (space and time) and ψ the mapping from our labels onto W. Then the group gets projected by this surjective, non-injective homomorphism (epimorphism) ψ into something representing the ordinary rules of Chess piece movement.

$\mathrm{Group}(\mathrm{V}) \rightarrow \mathrm{PieceMovement}(\mathrm{W})$

It's an homomorphism, because ψ after the group action composition gives the same as the piece movement addition after doing ψ first. More formally the mapping ψ between those two systems actually transfers a structure from an algebraic system to a set, and then by means of ψ one can define a structure on the set, and this finally makes ψ an isomorphism (if the mapping is one-to-one (injective) and onto (surjective)). The advantage is: we can treat the first with group theory, whereas the piece movement rules are not that nice, they are implicit and not explicit. It remains to be seen how the actual piece movement rules can now be modelled with our groups.

The idea of linear operators does carry over from vector spaces (that is Abelian) to our situation. Operator means to specify for all elements of the group another group element to become the image. And linear simply means that it is enough to specify the images for the generators, and then the images for all group elements can be constructed because of linearity A(abaa..) = A(a)A(b)A(a)A(a)A(...). Please don't confuse linear and homomorph mappings though.

Let's use the following lines on a rubik_cube_3: The resulting group then is ...

Perhaps our chosen connection can be characterized as a solution of an universal mapping problem, that is: given another connection (resulting in a connected group), then there exists a unique map such that the following diagram is commutative ...

11.3 Group Cohomology

11.3.1 Introduction

All those Platonic polyhedral groups are subgroups of SO(3) which has double cover SU(2). There are connections (see [89]) to so called icosians, quaternions, the E8- as well as even the Leech-lattice. Some people call the icosians the group algebra of A(5), and others treat the icosian ring as the 'Z-span of the root system of the non-crystallographic Coxeter group H_4'. The Binary Polyhedral Groups (as well called covering groups) for our three examples are $2.A(4) \cong SL(2,3)$, $2.S(4)$ and $2.A(5) \cong SL(2,5)$. We speak of a Double Binary Polyhedral Group when reflections are also included. And the universal cover has to do with an embedding into a simply-connected group.

Now (we follow [64]) let K, E, G be some groups, then the following short sequence:

$$1 \rightarrow K \underset{\alpha}{\rightarrow} E \underset{\beta}{\rightarrow} G \rightarrow 1 \tag{11.1}$$

is exact iff (if and only if) α is injective, β is surjective and Kernel(β) = Image(α), so β induces an isomorphism $G \cong \mathrm{E} / \mathrm{Im}(\alpha)$ and can be written $G \cong \mathrm{E} / \mathrm{K}$.

This is called a short exact sequence. We say E is an extension of G by K (in most cases if G is nonsolvable and K Abelian). The extension is split if there is a group homomorphism $\sigma : G \to E$ such that $\sigma \circ \beta = 1$, otherwise nonsplit. K is the extension, G is the quotient. E operates on K by conjugation and so we have a homomorphism $E \to \mathrm{Aut(K)}$. In case K is Abelian, it's in the kernel of this map and we have a homomorphism from G to Aut(K), whence K acquires the structure of a G-module. An extension is central if the extension kernel is in the center, and it's covering (an essential central extension in other words) if the kernel is in the derived group. A central extension is a covering group if the extension is maximal essential.

See [73] and [61] a representation group (Darstellungsgruppe) is a group E which is a central extension of G such that the following diagram (11.2) commutes:

$$\begin{tikzcd} E \arrow[r] \arrow[d] & \mathrm{GL(n,F)} \arrow[d] \\ G \arrow[r] & \mathrm{PGL(n,F)} \end{tikzcd} \qquad (11.2)$$

In case F = GF(q), we write GL(n,q) instead of GL(n,F). Pay attention to the fact that PSL(2,5) needs 6 (and not 5) labels as a permutation group (5 and infinity), and PSL(IsMatrixGroup,2,5) doesn't work.

E is an auxiliary object (in general not unique up to isomorphism) and lifts projective representations (for example over the complex numbers, the field $\mathbb{C}$) to linear ones. It's also called a universal covering group of G. In case G finite, a covering group exists and any such is a representation group, and if G is perfect then a covering group is unique (up to isomorphism). K is called the 'Schur multiplier' which can as well be defined as $\mathrm{H}^2(\mathrm{G},\mathrm{F}^x)$ where F^x denotes the multiplicative group of nonzero elements of some algebraically closed field F.

For example A(4), S(4) and A(5) have got the Schur multiplier Z(2).

$\mathrm{A}(4) \cong (\mathrm{Z}(2) \times \mathrm{Z}(2)) : \mathrm{Z}(3)$ and $\hat{E}_6 = 2.A(4)$ is actually (see [104]) $\mathrm{Q} : \mathrm{Z}(3) \cong \mathrm{SL}(2,3)$ and the preimage of A(4) in SU(2), so the stuff below becomes a short exact sequence:

$$1 \to \pm 1 \to \hat{E}_6 \to \mathrm{A}(4) \to 1 \qquad (11.3)$$

The perfect double cover 2.A(5) of A(5) is the icosian group and as well called the Binary Icosahedral group. See Chapter 15 for more information about the quaternions and icosians.

Now let's see if the Binary Polyhedral Groups are split or not:

```
gap>G:=AlternatingGroup(5);;
gap>d:=SchurCover(G);;
gap>i:=Image(IsomorphismPermGroup(d),d);;
gap>l:=NormalSubgroups(i);;
gap>GG:=i/l[2];;
gap>IsomorphismGroups(GG, G);
ok
gap>hom:=NaturalHomomorphismByNormalSubgroup(i,l[2]);;
gap>IsomorphismGroups(ImagesSource(hom),G);
ok
gap>FactorGroup(i,l[2]);;
gap>OneCocycles(i,l[2]);
... isSplitExtension := false
gap>sp25:=Sp(IsMatrixGroup,2,5);
SL(2,5)
gap>IsSimple(sp25);
false;
gap>IsSolvable(sp25);
false;
gap>RadicalGroup(sp25);;
gap>u:=SylowSubgroup(sp25,2);; Size(u);
8
gap>Size(Normalizer(sp25,u));
24
gap>Size(Centralizer(sp25,u));
2
```

Well, let's for now continue with the cover thing. When we start with the holonomy group G of the ico_stl_II example:

```
gap>AbelianInvariantsMultiplier(G);
[ 2, 2 ]
gap>SCG:=SchurCover(G);;
gap>Size(SCG)/Size(G);
4
```

It seems now there some deeper work is needed what we shouldn't postpone for too long. For a finite group G being polycyclic and solvable are the same. In that case the following is feasable (doubletetra example, other candidates are rhombic_dode, rubik_cube 2 and 3, costa, dcb_068, oct_stl_II etc.):

```
gap>Size(G);
648
gap>IsSolvable(G);
true;
gap>gp:=Image(IsomorphismPcGroup(G));;
gap>mats:=List( Pcgs(gp), x -> IdentityMat(1, GF(2)) );;
gap>M:=GModuleByMats(mats, GF(2));;
gap>TwoCohomology(gp,M);;
gap>li:=Extensions(gp, M);;
gap>List(li, ModuleOfExtension);;
gap>cli:=List(li, h -> Intersection(Centre(h), DerivedSubgroup(h)));;
gap>AbelianInvariants(Centre(li[4]));
[ 2 ]
gap>sc:=SchurCover(G);;
gap>iso:=IsomorphismGroups(li[2], sc);
fail
gap>iso:=IsomorphismGroups(li[4], sc);
ok
```

Alternatively we can use another package:

```
gap>LoadPackage("polycyclic");;
gap>gpc:=Image(IsomorphismPcpGroup(G));;
gap>mats:=List( Pcgs(gpc), x -> IdentityMat(1, GF(2)) );;
gap>cr:=CRRecordByMats(gpc,mats);;
gap>TwoCohomologyCR(cr);;
gap>ExtensionClassesCR(cr);;
```

Btw. in the context of algorithmic treatment considerations one often deals with polycyclic-by-finite groups, meaning it has a polycyclic subgroup of finite index.

The polyhedral groups and their binary counterparts have defining relations (Table 11.3).

graph	relations
tetrahedron	x^3 = y^3 = z^2 = xyz
cube/octahedron	x^4 = y^3 = z^2 = xyz
icosahedron/dodecahedron	x^5 = y^3 = z^2 = xyz

Table 11.3: *Relations of (Binary) Polyhedral Groups*

That's why 2.S(4) for example is called the order 48 Binary Octahedral (reflection) group {4,3,2}. Those relations equal -1 in the cover but 1 when projected to the polyhedral group. Working on directed edges, x can be looked at as the local rotation of faces, y as the dual rot (backwards) around vertices, and z^{-1} = xy is the edge flip then.

See [68] it's similar to central extensions of triangle groups like

```
< a,b,c,z with 1=[z,a]=[z,b]=[z,c]
      and a^1 = z^p, b^m = z^q, c^n = z^r, abc = z^s >
```

where z is some torsion generator.

11.3.2 Cohomology using Sylow subgroups

First of all we will use the 'cohomolo' package a little bit:

```
gap>LoadPackage("cohomolo");;
gap>G:=AlternatingGroup(5));;
gap>chr:=CHR(G,2);;
gap>SetInfoLevel(InfoCohomolo,1);
gap>SchurMultiplier(chr);
#I  Indices in the subgroup chain are: 5 3
[ 2 ]
gap>CoveringGroup(chr);;
```

Once a Schur cover is found, subsequent attempts to embed the result again into GL won't change anything. For now let's take a look (Table 11.4) at the Schur multipliers of the holonomy groups for our graph examples (we write [remedy] in case some helping measures would be needed for further accomplishments).

One has to test for all primes dividing the group order. For example we set G:=PSL(3,4), and using CHR(G,2) we get a multiplier [4, 4], and with CHR(G,3) we get [3], and G has a Schur cover (Z(12) × Z(4)).PSL(3,4).

The 3×4 torus, the hexa torus and M(24) all get [ ]. We speak of 'torsion-free', this is a property of a Riemannian connection. Otherwise it's called a Riemann-Cartan manifold.

The hexa_tor has got a Z(2) multiplier. $\forall n, n \geq 8: \mathrm{multiplier}(\mathrm{A(n)}) = \mathrm{Z}(2)$.

11.3.3 What _is_ a Schur cover?

We now present another constructive approach without the need of additional group-theoretic background. Remember that we mentioned some full (counter-clockwise) spin when introducing the holonomy groups and looking at $[a,b]^3$ or $(abc)^4$? Let's take a cube with group S(4) and a Chess piece of type A (Example 11.2). The Schur multiplier is a GF(2) G-module:

Example 11.2 Cube and Schur Cover

```
gap>G := Group(
  (11,10,12,13)(1,7,6,5)(2,16,17,18)(19,9,3,20)(23,8,15,22)(4,24,21,14),
  (1,2,3,4)(7,8,9,10)(19,17,6,21)(5,13,20,22)(23,18,12,24)(11,16,15,14)
);;
gap>SCG := Group(
  (1,7,6,5,25,31,30,29)
  (2,16,17,18,26,40,41,42)(3,20,19,9,27,44,43,33)(4,24,21,14,28,48,45,38)
  (8,15,46,47,32,39,22,23)(10,36,37,35,34,12,13,11),
  (1,2,3,4,25,26,27,28)
  (7,8,33,10,31,32,9,34)(6,45,43,41,30,21,19,17)(5,13,44,22,29,37,20,46)
  (16,15,38,11,40,39,14,35)(18,12,48,23,42,36,24,47)
);;
```

We start with the "b" generator's cycle (1,2,3,4) and double the labels (+24) since after a^4 we have got the spinned situation (for example the label 1 stands for a coordinate system with the "a" axis from face with id 0 to a face with id 1, and the "b" axis rotated counter-clockwise thereby going from 0 to 2). Then we do the "a" generator for all already occured labels. Always we use the standard label of the original group for the initial occurence, and for the first four cycles of the "a" generator there are always fresh labels to use. Well, so then we continue with the "b" generator, and there finally the question arises if to use the label 9 or 33. The situation did occur already, 9 = (bbaaa) 1, and we are working on the label 7 = (a) 1, and it's now at (9 or 33) = (bb) 7. The loop goes (cccddabb) and we shrink it to a point (also called collapsing or contracting) by simplyfying (inserting identities such as ac or bd (to break into atomic rots), using rot = abc = cda along cycles, rota = brot to move the rots to one end; aabb = rotˆ2 or other shortcuts), find that we have rotˆ4 and use 33. Actually the first time an already used label is found to be not rotated is at 15 because $\mathrm{dcdaba} \; = \; \mathrm{drotba} = \mathrm{dcrota} = \mathrm{dcbrot} = \mathrm{rot\hat{}} - 1\mathrm{rot} = 1$.

Depending on if you shrink to the left or the right you get rot^4 or rot^{-4}. Only after two full rotations the original situation is reached again. In other words, a full clockwise rotation gives the same as a full counter-clockwise rotation. It seems this is because it's not possible to store more information (than these two states) in such a short cycle (compare Section 11.4 where we speak about the finiteness of the morphing groups).

graph	**multiplier**
tetrahedron	[2]
deltoid / delt_trunc	[2]
tetra_stl_II	[2, 2]
cube/octahedron	[2]
icosahedron/dodecahedron	[2]
icosa var_1	[remedy]
doubletetrahedron	[2]
tetra_/(i_s_3 / i_s_3_1)	[2, 2]
tetra_i_s_3_2	[2, 2, 2]
cube_triang_I	[2]
cube_triang_II	[2] according to the literature
bipyramid_(penta / hepta / octa)	[2]
oct_stl_I	[4]
tetra_reticul	[2] [3]
tetra_reticul_big	[8]
tetra_reticul_tall	[2] [5]
oct_stl_II	[2, 2, 2]
M_24	[]
L2_7 / S5	[2]
A6	[2][3]
doublecube	[2, 2, 2, 2]
doublecube hole_10	[2, 2, 2]
doublecube hole_0_10	[2]
doublecube hole_8	[remedy]
doublecube hole_2_8	[2, 2, 2, 2]
doublecube hole_2_3_6_8	[ ]
doublecube hole_6_8	[remedy]
doublecube hole_0_6_8	[2, 2]
doublecube hole_3_8	[2, 2, 2, 2] [3]
2x3 rectangle	[2, 2]
triplecube	[2, 2, 2]
ico_stl_I	[2]
ico_stl_II	[2, 2]
torus_(3x4 / hexa_3x5)	[]
trapezo	[2, 2, 2]
quad_star	[2, 2][remedy]
rubik_cube_2	[2, 2] [3, 3]
rubik_cube_3	[2][remedy]
rubik_cube_3 var_1	[remedy]
rhombic_dodeca	[2, 2]
zonotope_5	[2]
yabi / monkgau	[remedy]
costa	[4, 4, 4, 4, 4, 4, 2]
quasi_0	[2]
quasi_1	[remedy]
quasi_(tria_0 / hepta_0)	[2]
icositetra	[2, 2, 2]
doubledode	[remedy]
dode_tor_6	[remedy]
hexa_tor	[2] according to the literature

Table 11.4: *Schur multiplier*

The next easy example is the tetrahedron, you will manage for yourself. We will give you some hints: you will have to insert on the line, that is $aaa = (bb\hat{} - 1)aaa = b(aa\hat{} - 1)b\hat{} - 1aaa = rot\hat{} - 1a\hat{} - 1b\hat{} - 1aaa$ etc., and it's a good idea to move negative powers of rot to one side (left) and positive powers to the other (right). Besides you will make use of $\text{dual\_rot}^{\text{local\_dual\_valence}} = -1 = \text{rot}^{\text{poly}}$ finally. Come on, just try and feel the spirit.

Well, once you have understood the construction it's quite easy and only a technical task to teach it to the computer. Interesting to visualize how such a loop gets moved on the graph while collecting information about the torsion thereby.

By the way, rota = brot can be equivalently formulated as: the two automorphisms "a" and "b" (of some algebra) are similar, there exists an intertwining automorphism called "rot".

```
gap>sc:=SchurCover(PGL(2,3));;
gap>IsomorphismGroups(sc, GL(2,3));
ok
gap>IsomorphismGroups(sc, SCG);
fail
gap>IsPerfect(G);
false
```

As we know the desired group is not uniquely determined, so no problem with that output above either.

11.3.4 Schur covers and lifting a quotient

Let's take a look at the following lovely little script (see the GAP source code in the file lib/schur.gi) for computing a finite presentation of a representation group (Darstellungsgruppe, Schur cover) of a finitely presented group, using the Hopf formula. Simply take a finite presentation F/R for a group G and compute a presentation of one of G's representation groups. This is done by assembling a presentation for F/[R,F] and then finding a generating set for a complement C/[R,F] for the intersection of R and [F,F] in R/[R,F] (all possible covers would be achieved by computing all complements).

```
gap>DarstellungsgruppeFP := function( G )
  local g, i, m, n, r, D, I, M, M2, fgens, rels, gens, Drels;

  fgens:=FreeGeneratorsOfFpGroup(G);
  rels:=RelatorsOfFpGroup(G);
  n := Length( fgens );
  m := Length( rels );

  D := FreeGroup( n+m );
  gens:=GeneratorsOfGroup(D);
  Drels := [];
  for i in [1..m] do
     r := rels[i];
     Add(Drels, MappedWord( r, fgens, gens{[1..n]} ) / gens[n+i] );
  od;
  for g in gens{[1..n]} do
     for r in gens{[n+1..n+m]} do
        Add( Drels, Comm( r, g ) );
     od;
  od;

  M := [];
  for r in rels do
     Add( M, List( fgens, g->ExponentSumWord( r, g ) ) );
  od;

  M{[1..m]}{[n+1..n+m]} := IdentityMat(m);
  M := HermiteNormalFormIntegerMat( M );           ❶[1]
  M:=Filtered(M,i->not IsZero(i));

  r := 1; i := 1;
  while r <= m and i <= n do
     while i <= n and M[r][i] = 0 do
```

```
      i := i+1;
   od;
   if i <= n then  r := r+1; fi;
od;
r := r-1;

if r > 0 then
   M2 := M{[1..r]}{[n+1..n+m]};
   M2 := HermiteNormalFormIntegerMat( M2 );      ❷[1]
   M2:=Filtered(M2,i->not IsZero(i));
   for i in [1..Length(M2)] do
      Add(Drels,LinearCombinationPcgs(gens{[n+1..n+m]},M2[i]));
   od;
fi;

D:=D/Drels;
return D;
 end;
```

❶, ❷ here `HermiteNormalFormIntegerMat` is used

Now let G denote the holonomy group of the double-tetrahedron example:

```
gap>gens:=GeneratorsOfGroup(G);;
gap>iso:=IsomorphismFpGroupByGenerators(G, gens);;
gap>G_fp:=Image(iso);;
gap>dar:=DarstellungsgruppeFP(G_fp);
<fp group on the generators [ f1, f2, f3, f4, f5, f6, f7, f8 ]>
gap># (now K:=SimplifiedFpGroup(dar);; would be an option)
gap>P:=PresentationFpGroup(dar);;
gap>TzOptions(P).protected:=2;;
gap>TzOptions(P).printLevel:=2;;
gap>TzOptions(P).loopLimit:=10;;
gap>v:=GeneratorsOfPresentation(P);;
gap>TzEliminate(P,v[3]);;
gap>SimplifyPresentation(P);;
gap>K:=FpGroupPresentation(P);
<fp group on the generators [ f1, f2 ]>
gap>is:=IsomorphismPermGroup(K);;
gap>K_p:=Image(is);;
gap>SmallerDegreePermutationRepresentation(K_p);;
IdentityMapping( <permutation group of size 1296 with 2 generators> )
gap>NrMovedPoints(K_p);
 432
```

That's too much, and for bigger graphs the isomorphism to a permutation group is anyway not feasable anymore. Can we find a better solution?

```
gap>tom:=TableOfMarks(K_p);;
gap>ctl:=CharacterTable(K_p);;
gap>trueperms:=PermCharsTom(ctl,tom);;
```

There are for example some of degree 36, but how to go from such a character to the corresponding representation? And while they are not just possible candidates for characters, still that's only a necessary (and not sufficient) condition for s.th. faithful. Actually the kernel of a character are all group elements where the character has got the same value as for the identity element, and the corresponding representation is faithful if and only if the kernel is trivial (so in the example above 48 could be the minimal degree).

By the way, here you get s.th. isomorph to G on just 9 points (but somehow we still prefer the 18 points, probably because then we need only two generators):

```
gap>blocks:=Blocks(G, MovedPoints(G));
[ [ 1, 3 ], [ 2, 4 ], [ 5, 10 ], [ 6, 11 ], [ 7, 12 ], [ 8, 13 ], [ 9, 14 ],
  [ 15, 16 ], [ 17, 18 ] ]
gap>blockhom:=ActionHomomorphism(G,blocks,OnSets);;
gap>Gb:=Image(blockhom);;
gap>IsomorphismGroups(G,Gb);;
gap>Length(GeneratorsOfGroup(Gb));
 7
```

Actually the question arises what is the minimal degree of a faithful permutation representation for our cover. It is the index of the largest core-free subgroup. Well, core-free is definitely not a synonym for Abelian. Let us mention that a permutation representation of degree n is actually a homomorphism into S(n) and leave this minimality problem for the moment.

For bigger examples the approach above is not feasable. We will now very closely follow [70] (more often than not by verbatim copy/paste whole passages, but we must be explicit in our construction for a wide, unexperienced audience) and lift our permutation representation for G of degree 18 to a cover (on 144 labels). Perhaps (see the mentioned paper for the actual state of the art) it is not completely sophisticated to work with some arbitrary Darstellungsgruppe, but for the moment this will just suffice:

```
gap>h:=GroupHomomorphismByImages(K,G,GeneratorsOfGroup(K),GeneratorsOfGroup(G));
[ f1, f2 ] -> [ (1,2,3,4)(5,15,11,8)(6,13,10,16)(7,17,14)(9,12,18),
  (1,5,6)(2,7,8,9)(3,10,11)(4,12,13,14)(15,17,16,18) ]
gap>Size(Image(MaximalAbelianQuotient(K)));
2
gap>DerivedSubgroup(K);
Group([ f1^-2, f2*f1^-1 ])
gap>Size(last)
648
gap>Size(Kernel(h));
2
```

The kernel(h) must be $\leq$ K′, that means the largest Abelian quotient of the cover K $\cong$ that of G . Now let s be the subgroup which is the preimage of a point stabilizer:

```
gap>s:=PreImage(h,Stabilizer(G,1));; Index(K,s);
18
gap>h2:=IsomorphismFpGroup(s);;
gap>q:=Range(h2);;
```

Next, we compute the permutation images of the (new) generator preimages under the epimorphism onto G and construct the corresponding epimorphism from the new fp group onto the point stabilizer in G.

```
gap>gens:=List(GeneratorsOfGroup(q),i->Image(h,PreImagesRepresentative(h2,i)));
[ (7,14,17)(9,18,12), (2,9,8,7)(4,14,13,12)(15,18,16,17) ]
gap>h3:=GroupHomomorphismByImages(q,Subgroup(G,gens),GeneratorsOfGroup(q),gens);
#I  CosetTableFromGensAndRels called:
#I    2 1 1 2
#I  CosetTableFromGensAndRels called:
#I    2 1 1 2
[ F1, F2 ] -> [ (7,14,17)(9,18,12), (2,9,8,7)(4,14,13,12)(15,18,16,17) ]
gap>o:=Orbits(Range(h3),[1..18]);
[ [ 1 ], [ 2, 9, 8, 13, 7, 18, 16, 12, 4, 15, 14, 17 ], [ 3 ],
 [ 5 ], [ 6 ], [ 10 ], [ 11 ] ]
```

Taking the orbit of length 12 now would give the inferior result on 432 points from above.

```
gap>t:=Stabilizer(Image(h3),o[1][1]);
Group([ (7,14,17)(9,18,12), (2,9,8,7)(4,14,13,12)(15,18,16,17) ])
gap>DisplayCompositionSeries(t);;
gap>tp:=PreImage(h3,t);; Index(q,tp);
1
```

We now compute the epimorphism from tp onto its commutator factor group (by first rewriting the presentation to one of tp and then Abelianizing the presentation, the resulting Abelian quotient of size 8 is represented as a pc group).

```
gap>maxab:=MaximalAbelianQuotient(tp);; Size(Image(maxab));
8
```

Let's pull this quotient back to a subgroup of the finitely presented group K (of index 18). For this we need a generating set for this subgroup (which is obtained by taking the primary generators of an augmented coset table).

```
gap>U:=PreImage(h2,tp);; Index(K,U);
18
gap>Ugens:=GeneratorsOfGroup(U);; Length(Ugens);
4
gap>Umax:=RestrictedMapping(h2,U)*maxab;;
```

We want to induce this representation to K. Now first of all the ugly way, involving the computation of a stabilizer chain for the permutation image, this won't be ok for bigger groups. The representation below on the cosets is faithful if and only if the core of ke in K (that is the intersection of all K-conjugates of ke) is trivial.

```
gap>ke:=Kernel(Umax);; Index(K,ke);
144
gap>Core(K,ke);
Group(())
gap>tab:=CosetTable(K,ke);;
gap>tab:=List(tab{[1,3..Length(tab)-1]},PermList);;
gap>kk:=Group(tab);; Size(kk); Size(G);
1296
648
```

And now instead here comes the better way how to do it. By the Krasner-Kaloujnine embedding theorem, the induced permutation representation goes in a wreath product of the images of the original representation with the permutation representation on the cosets. DefiningQuotientHomomorphism of a subgroup is not necessarily the action on the cosets, but only some homomorphism whose kernel is contained in the subgroup. In this particular case it is this particular homomorphism.

```
gap>Ucosrep:=DefiningQuotientHomomorphism(U);;
gap>perms:=KuKGenerators(K,Ucosrep,Umax);;
gap>NrMovedPoints(perms);
144
```

Now we could first of all use IsomorphismSimplifiedFpGroup(K); to find redundant generators within the perms, but for the moment we just take them all. The use of straight line program elements might be overkill in the concrete example but will be appropriate in bigger cases.

```
gap>p3:=StraightLineProgGens(perms);;
gap>P:=Group(p3);;
gap>StabChainOptions(P).random:=1;; Size(P);
1296
gap>bas:=BaseStabChain(StabChainMutable(P));;
gap>Length(bas);
3
```

The computed stabilizer chain provides us with a base for the group. Knowledge of this base will speed up comparisons of straight line program elements enormously (we only have to test equality on the 3 points of the base instead of the 144 points of the domain). For the further calculations we therefore create the straight line program generators anew, this time with a base. We also create the permutation group P anew and store its size.

```
gap>p3:=StraightLineProgGens(perms,bas);; P:=Group(p3);
gap>SetSize(P,Size(G)*2);;
```

So finally we have the result for our double-tetrahedron example (Example 11.3).

Example 11.3 DoubleTetrahedron and Schur Cover

```
gap>GeneratorsOfGroup(G);
[ (1,2,3,4)(5,15,11,8)(6,13,10,16)(7,17,14)(9,12,18),
  (1,5,6)(2,7,8,9)(3,10,11)(4,12,13,14)(15,17,16,18)
]
gap>GeneratorsOfGroup(P);
[ (1,9,41,17,2,10,42,18)(3,11,43,19,4,12,44,20)(5,13,45,21,6,14,46,22) ...
,
  (1,25,33,5,29,37,3,27,35,7,31,39,2,26,34,6,30,38,4,28,36,8,32,40) ...
]
```

You can see that the original 4-cycles are now looping twice (and all this in 4 copies in parallel), and the old 3-cycles go 2^3-fold around.

Finally let's look at the kernel of the lift:

```
gap>RelatorsOfFpGroup(dar)[1];
f1^-2*f2^-1*f1^-2*f2^-1*f3^-1
gap>elm:=(P.1^-2*P.2^-1*P.1^-2*P.2^-1); Order(elm);
2
gap>S:=SubgroupNC(P,[elm]);;
gap>N:=SolvableNormalClosurePermGroup(P,S);;
```

N is indeed the whole kernel - otherwise we would have had to add further elements.

```
gap>IsElementaryAbelian(N);
true
gap>pcgs:=Pcgs(N);;
gap>mats:=LinearActionLayer(P,pcgs);
gap>module:=GModuleByMats(mats,GF(2));
rec( field := GF(2), isMTXModule := true, dimension := 1,
  generators := [ <an immutable 1x1 matrix over GF2>,
             <an immutable 1x1 matrix over GF2> ] )
gap>MTX.IsIrreducible(module);
true
```

Now looking at similar graphs, we recognize that the 5-bipyramid and 7-bipyramid as well behave alike. And the 6-bipyramid shows familiar behavior: the old 4-cycles loop twice again (in 2 copies), and the original 6-cycles go 2^2-fold. The 8-bipyramid then shows only 16-cycles.

Let's investigate the results for oct_stl_I (small tetrahedron reticulated) type A holonomy group (Example 11.4).

Example 11.4 Tetrahedron Reticulated and Schur Cover

```
gap>GeneratorsOfGroup(G);
[ (1,2,3)(4,19,13,20,21,22)(5,17,23,24,25,26)(6,27,28) ...
,
  (1,4,5,6,7,8)(2,9,10,11,12,13)(3,14,15,16,17,18)(19,37,36) ...
]
gap>GeneratorsOfGroup(P);
[ (1,9,17,8,16,24,2,10,18,7,15,23)(3,11,19,5,13,21,4,12,20,6,14,22) ...
,
  (1,25,89,217,116,35)(2,26,90,218,115,36)(3,27,91,219,113,34) ...
]
```

Again it's 2^3-fold sheets, but [4] as kernel. Now the result for dcb_hole_0_10:

Example 11.5 DoubleCube with two holes and Schur Cover

```
gap>G;
Group([ (1,2,3,4)(5,6,7,8), (1,5)(2,6)(3,7)(4,8) ])
gap>P;
Group([ (1,2,6,3)(4,8,12,10)(5,7,13,9)(11,14,16,15),
        (1,4,11,5)(2,7,14,8)(3,9,15,10)(6,12,16,13) ])
```

Finally we present the result for the deltoid (stellated tetrahedron) (see Example 11.6).

Example 11.6 Tetrahedron Stellated and Schur Cover

```
gap>G;
Group([ (1,2,3,4,5,6)(7,28,25)(8,26,29,20,17,30)(9,23,16) ...
,
        (1,7,8,9,10,11)(2,12,13)(3,14,15,16,17,18)(4,19,20) ...
])
gap>P;
Group([ (1,2,6,18,9,3)(4,12,35,79,37,13)(5,15,43,67,46,16) ...
,
        (1,4,14,42,17,5)(2,7,23,65,26,8)(3,10,31,82,34,11) ...
]);
```

Dealing with the big reticulated tetrahedron we have Z(6), and it is realized with 6-cycles only (with a total of 12-fold sheets) whilst we luckily had no need for s.th. iterated. The bigger reticulated tetrahedron gets a 2^4-fold cover with cycles of length 24 (in 2 copies) and kernel Z(8). The tall reticulation then again uses only 6-cycles for a total of 20-fold cover and kernel Z(10).

The L2_7 and S5 covers get a doubled set of labels and correct cover group, whereby the orbit cycle lenghts are the same as those of the original group. Perhaps that's why while having a non-trivial Schur multiplier they are nevertheless still called Riemannian instead of Cartanian manifolds?!

In case of the ico_stl_I (and delt_trunc etc.) graph the above procedure somehow multiplies the label set without actually changing the group towards a proper cover, it seems this happens because the Core(K,ke) is nontrivial and we must take care of this.

11.3.5 Results

Here (Table 11.5) are some results for various graphs (multiplier, sheets, number of conjugacy classes of a Schur cover (what is the same as the number of Irrs), and the lists of degrees of the absolutely irreducible complex characters), given by

```
gap>ctl:=CharacterTable(P);
gap>Length(Irr(ctl));
gap>CharacterDegrees(ctl);
```

At the moment we restricted ourselves to genus 0 graphs (because there is only the 'shrinking to a point', on other graphs we have additional situations to arrive at).

It's not always some SL(2,q) (Galois's last dream doesn't allow this for A(432) and reasonably small q). Well, |SL(2,q)| = |PGL(2,q)| = q(q^2-1), we can use this as a quick check.

We might speculate that things like invariant metric or bilinear forms in general can now be retrieved as module isomorphisms from some module M over GF(q) defined by G to its dual module (see [67]).

11.4 Morphing Group

Here a goup theoretical treatment of Rubik's Cube variants could take place. It's all about cutting and reglueing along edges between vertices.

It's important to realize that the resulting groups are actually finite. Let's take a torus and morph again and again. Thereby we store some windings in the graph, and of course this can't go to infinity. There is a point where we reach a graph isomorph to s.th. already constructed (a physical model would likely break during the experiment).

The standard Rubik's move is built out of three subsequent atomic moves, such that finally the resulting graph doesn't differ from the original one. But see Section 7.24 we can (by allowing more general moves) construct new graphs, for example also situations can arise where faces might be connected by more than one neighbouring edge. In order to detect isomorphic variants the standard holonomy groups were constructed on all graphs, and then the lenghts of the orbits were compared.

How many topologically different graphs can be created by atomic moves (up to symmetry)? The number is too large to allow a construction of all graphs, for example in case of the icosahedron one gets already about 250 morphed states after only one (possibly non-straight) move (whereby crossings and touchings of the cutting lines are even left out), and the procedure can then be applied subsequently to all these graphs (and so on), quickly reaching several thousands of variants.

graph	multiplier	#sheets	#Irrs	degrees
tetrahedron	Z(2)	2	7	[1,3][2,3][3,1]
cube/octahedron	Z(2)	2	8	[1,2][2,3][3,2][4,1]
icosahedron / dodecahedron	Z(2)	2	9	[1,1][2,2][3,2][4,2] [5,1][6,1]
icosa var_1				
tetra_stl_I (deltoid)	Z(2)	4	21	[1,6][2,9][3,2][4,3] [6,1]
delt_trunc	Z(2)	6		
tetra_stl_II	Z(2)^{2}			
doubletetrahedron	Z(2)	8	26	[1,2][2,3][3,2][4,1] [6,8][8,6][12,4]
cube_triang_I (bipyramid_hexa)	Z(2)	4	26	[1,2][2,3][3,2][4,1] [6,8][8,6][12,4]
cube_triang_II	Z(2)			
bipyramid_penta	Z(2)	8	50	[1,2][2,3][3,2][4,1] [6,16][8,12][12,8][24,6]
bipyramid_hepta	Z(2)	8	84	[1,2][2,3][3,2][4,1] [6,24][8,18][12,12][24,22]
bipyramid_octa	Z(2)	8	18	[1,2][2,3][3,6][4,1] [6,3][8,3]
oct_stl_I	Z(4)	8	30	[1,6][2,6][3,2][4,12] [6,4]
tetra_reticul	Z(6)	12	54	[1,6][2,9][3,14][4,3] [6,17][9,4][12,1]
tetra_reticul_big	Z(8)	16	68	[1,6][2,6][3,2][4,12] [6,12][8,24][12,6]
tetra_reticul_tall	Z(10)	20	90	[1,6][2,6][3,2][5,24] [6,16][10,24][12,4][15,8]
oct_stl_II				
L2_7	Z(2)	2	11	[1,1][3,2][4,2][[6,3] [7,1][8,2]
M_24	()	1	26	[1,1][23,1][45,2][231,2] [252,1][253,1][483,1][770,2] [990,2][1035,3][1265,1] [1771,1][2024,1][2277,1] [3312,1][3520,1][5313,1] [5544,1][5796,1][10395,1]
doublecube				
doublecube h_10				
doublecube_h_0_10	Z(2)	2	10	[1,8][2,2]
2x3 rectangle	Z(2)^{2}	8	37	[1,6][2,6][3,2][4,15] [6,4][8,3][12,1]
triplecube				
ico_stl_I	Z(2)	2		
ico_stl_II				
torus_3x4	()	1	12	[1,12]
torus_hexa_3	()	1	15	[1,15]
S5	Z(2)	2	12	[1,2][4,5][5,2][6,3]
quad_star				
rubik_cube_[2,3]				
rhombic_dodeca	Z(2)^{2}	6	26	[1,4][2,9][4,2][6,4] [9,4][12,2][18,1]
zonotope_5				
costa	Z(2) × Z(4)^{6}			
quasi_0	Z(2)	50	145	[1,20][2,85][4,40]
quasi_tria_0	Z(2)	2	15	[1,12][2,3]
quasi_hepta_0	Z(2)	98	329	[1,28][2,175][4,126]

Table 11.5: *Holonomy Group Schur Covers*

Instead one should restrict the investigation to straight cut/glue lines. This is in analogy to the holonomy groups, where also straight lines play a crucial role. Now it's along vertices instead of faces, but again one has got the possibility to have one line being rotated against another one etc. It makes sense to allow touchings and crossings, making the cut/glue process somewhat more difficult, it must be done locally on the fly and can't be done globally in one big action. A cut/glue line is considered closed when the intermediate triangle hole (which is created initially when starting the process, together with another hole being connected to the graph over a looping edge, and two of the edges of the triangle hole were formerly connected, and the third edge of the triangle hole was connected to the edge which is now looping around the other hole) is filled again (can be done along any of the three directed sides of the triangle hole). The group then consists of the movement (rotation and indirect translation) of the origin (where the cut/glue starts), together with the action on the several variants of graphs, and last not least the action on the contents of the faces (color etc.). Besides one could generalize this to also take the relative orientation of the faces into account (for example by using a texture as in the sliding puzzle game instead of only a simple coloring). The described group is a generalization of the standard Rubik cube group treatment (and doesn't rely on the possibility to assign absolute labels globally).

[To be continued ...]

Let's now focus on the standard permutation groups on fixed labels generated by pairs of the form twist/untwist along straight lines (without crossings or touchings) and without the creation of new graphs. It's already known that (yes, you probably already expected this goal) on the rubicon (icosahedron accompanied by Rubik - like moves on vertices) thereby we get the sporadic group M12 (Mathieu group) (see [89] and the cross groups in [152]) (ExportData -rf and -rg options):

graph	distIE	edge labels	face labels
tetrahedron	1		Z(2) × Z(2)
cube	1		A(5)
oct_stl_I	1		Z(2) × A(16)
oct_stl_II	1		A(48)
dodecahedron	1		A(12)
dodecahedron	2		M(12)
icosahedron	1		A(20)
ico_stl_I	1		Z(2) × A(60)
ico_stl_II	2		A(60)
rubik_cube_3_0	2		A(54)
rubik_cube_3_1	2		A(54)
dode_tor_6	2		Z(2) × A(58)
hexa_tor	1		Z(2) × A(72)

Table 11.6: *Cross Groups on Polyhedra*

Following the GAP example session about the Rubik's magic cube more aspects could be investigated.

11.5 Galois Groups

There exist algorithms to decompose any polynomial over $\mathbb{Z}$ into a product of polynomials over $\mathbb{Z}$ all of which are irreducible over $\mathbb{Q}$. If a polynomial is not irreducible, it's not always easy to find its Galois group in terms of the Galois groups of its irreducible factors, so we focus on (monic) irreducible polynomials. The irreducible parts of the graph's adjacency matrix's characteristic polynomial can be used to compute the Galois group of a corresponding field extension.

$$\text{Galois}(f) \text{ transitive} \Leftrightarrow f \text{ irreducible}$$

The study of an arbitrary group can be reduced to that of transitive groups (as a (sub)direct product), and imprimitive groups are built (embedded in the wreath product) from primitive components (not uniquely determined). See [71] for more information about transitive permutation groups.

The following polynomial of the 'quasi_1' example delivers a wreath product $S(6) \wr C(2)$:

```
gap>x:=Indeterminate(Rationals);;
gap>f:=-41 - 100*x + 284*x^2 + 694*x^3 + 40*x^4 - 661*x^5 - 258*x^6 +
      218*x^7 + 119*x^8 - 27*x^9 - 19*x^10 + x^11 + x^12;
gap>GaloisType(f);
299
gap>TransitiveGroup(12,299);
[S(6)^2]2=S(6)wr2
```

KASH version:

```
kash% Galois(f);
"[S(6)^2]2"
kash% Discriminant(f);
364686337094564303371953125
```

The signature (the number of real roots) for the polynomials in question equals the polynomial's degree. In Table 11.7) follow some general results:

graph	polynom	group
ico_stl_I	$4 + 6*x - 5*x^{2} - 2*x^{3} + x^{4}$	D(4)
quad_star	$-8 + 5*x + 14*x^{2} - 3*x^{3} - 4*x^{4} + x^{5}$	S(5)
fancy	$-2 + 69*x + 118*x^{2} - 134*x^{3} - 234*x^{4} + 41*x^{5} + 127*x^{6} + 11*x^{7} - 20*x^{8} - 2*x^{9} + x^{10}$	S(10)
rubik_cube_3 variant 1	$-3244 + 14696*x^{2} - 22543*x^{4} + 14369*x^{6} - 4301*x^{8} + 619*x^{10} - 41*x^{12} + x^{14}$	[2^7]S(7)
quasi variant 1	$-24 + 83*x^{2} - 20*x^{4} + x^{6}$	2S_4(6)
quasi variant 1	$-41 - 100*x + 284*^{2} + 694*x^{3} + 40*x^{4} - 661*x^{5} - 258*x^{6} + 218*x^{7} + 119*x^{8} - 27*x^{9} - 19*x^{10} + x^{11} + x^{12}$	[S(6)^2]2
m_24	$6 - 118*x + 48*x^{2} + 512*x^{3} + 29*x^{4} - 694*x^{5} - 273*x^{6} + 276*x^{7} + 160*x^{8} - 26*x^{9} - 24*x^{10} + x^{12}$	S(12)
rubik_cube_3	$12 + 111*x - 306*x^{2} - 2002*x^{3} + 1484*x^{4} + 10231*x^{5} - 83*x^{6} - 19231*x^{7} - 5886*x^{8} + 14263*x^{9} + 5906*x^{10} - 4936*x^{11} - 2269*x^{12} + 837*x^{13} + 403*x^{14} - 67*x^{15} - 33*x^{16} + 2*x^{17} + x^{18}$	T18_983
quad_hex	$-1 + 9*x + 127*x^{2} - 492*x^{3} - 3993*x^{4} + 1949*x^{5} + 27685*x^{6} + 4778*x^{7} - 67921*x^{8} - 13341*x^{9} + 88402*x^{10} + 6659*x^{11} - 68454*x^{12} + 5576*x^{13} + 31863*x^{14} - 7655*x^{15} - 8307*x^{16} + 3451*x^{17} + 910*x^{18} - 700*x^{19} + 41*x^{20} + 50*x^{21} - 13*x^{22} + x^{23}$	T23_7

Table 11.7: *Galois Groups*

11.6 ToDo

- parts of exporting data (permutations etc.) could be delegated to GAP in case the respective funtionality is available
- inverse problem: a group given, what is the smallest genus of a graph having the group as holonomy or automorphism group, and recontruct the graph(s)
- the surfaces may change in time instead of being stable, create the holonomy group on a simultaneously morphing graph, and relate the result to the original group from the static graph, and as well compare to the holonomy groups coming from morphed but static states
- the icosahedron is dual to dodecahedron, and on the other hand the term dual (dual space) is used for the <brak - ket> Hilbert product thing is there a connection?
- finish the implementation of -q -s 2 for boundary graphs (that is, expressing rot in the same labels as used for the vector's)

- perhaps the Loyd's 15 sliding puzzle could be modelled using the holonomy groups with the reflection at a boundary (instead of any groupoid)
- holonomy groups for all types of Chess pieces (not just type A)
- is a move of a type D piece invertible (how to define the holonomy group for it)?
- rewrite the methods creating orbits etc. to make use of multi-core architectures

Chapter 12

Representation Theory

12.1 Group Rings and Algebras

The group algebra is any of various constructions to assign to a group a ring or algebra over a field, such that the group multiplication induces the multiplication in the ring or algebra.

The group ring or group algebra (notation K[G], or sometimes just KG) allows (linear) representations of a group G over a field K to be treated as modules. It can be described as the vector space over K with basis the elements g of G, and ring multiplication the group operation in G extended by bilinearity to the whole space. The structure of K[G] as an associative algebra over K follows when we apply the distributive law and K-linearity.

The general isomorphism problem is to investigate when KG = KH implicates $G \cong H$

According to [67] methods are more highly developed for representations over finite fields, and there might arise some formidable problems when computing with representations over fields of characteristic zero, and over algebraic number fields in particular. When K = $\mathbb{Z}$ we speak of an integral representation.

Let p = char(K). If p = 0 or $p \nmid |G|$ (coprime, no common factor other than 1 and -1) it's the ordinary case (as in invariant theory). In the modular case (p > 0 and $p \mid |G|$) the group of units might happen to be not Abelian anymore in general.

Actually according to Maschke's theorem (see [74]) the group algebra KG of a finite group G is semisimple if and only if we have the ordinary case, and then reducibility and decomposability are the same.

For example over K = $\mathbb{R}$ or $\mathbb{C}$; the group algebra of a finite group is semisimple, and one can concentrate on irreducible modules. If over the complex numbers $\mathbb{C}$ there exists a faithful irreducible $\mathbb{C}$ G-module, then the centre of G is cyclic.

A simple algebra has no other (non-trivial) ideals besides itself. In case the maximal solvable ideal (the radical) vanishes, the algebra is called semisimple and is the direct sum of simple algebras. The quotient KG / Jacobson radical(KG) is always semisimple.

Normal subgroups in group theory, ideals in Lie algebras and two sided ideals in ring theory all arise as kernels of homomorphisms.

A K-algebra has finite representation type if it admits only finitely many non-isomorphic indecomposable modules. A group algebra over a field of positive characteristic has finite representation type if and only if its p-Sylow subgroups are cyclic. An algebra of infinite representation type is either tame or wild. In the former case all but finitely many indecomposable KG-modules of a given dimension can be parametrized by essentially one parameter of the base field. But if the algebra is wild, then the classification of the indecomposables is considered hopeless. Since most algebras are of wild representation type, one has to find a new way of describing indecomposable modules. This is accomplished by the so-called Auslander-Reiten quiver theory.

Whilst we cannot give a full overview but must restrict ourselves to the simple cases, we will now use the LAGUNA package, which extends GAP for computations with group rings and especially modular group algebras of finite p-groups (see [79]).

```
gap>G:=SymmetricGroup(3);
gap>K:=GF(2);
gap>KG:=GroupRing(K,G);
<algebra-with-one over GF(2), with 2 generators>
```

```
gap>IsPModularGroupAlgebra(KG);
false
```

If the order of G is not a power of p, G is not a p-finite group, and so we don't have a p-modular group algebra.

```
gap>IsUnit(One(KG));
true
gap>U:=Units(KG);
<group with 3 generators>
```

The last command is not handled by LAGUNA, but the present version will speed up only checking whether an element is a unit and computing its inverse only for those units of KG which are p-elements.

```
gap>Size(last);
12
gap>List(U,Order);
[ 1, 2, 6, 6, 2, 2, 2, 2, 2, 3, 3, 2 ]
gap>IsomorphismPcGroup(U);
[ (Z(2)^0)*(1,3), (Z(2)^0)*(1,3,2),
(Z(2)^0)*(2,3)+(Z(2)^0)*(1,2)+(Z(2)^0)*
    (1,2,3)+(Z(2)^0)*(1,3,2)+(Z(2)^0)*(1,3) ] -> [ f2, f3^2, f1 ]
gap>Image(last);
Group([ f1, f2, f3 ])
```

Now let's take another example. Here at the moment the group of units is out of reach already, but we can still display other useful features.

```
gap>G:=SL(2,5);;
gap>K:=GF(5);;
gap>KG:=GroupRing(K,G);;
gap>RadicalOfAlgebra(KG);
<algebra of dimension 65 over GF(2)>
```

The last command is from GAP (and for p-modular group algebras it will be replaced by a faster method from LAGUNA). The next series of command also is based on LAGUNA methods for Lie algebras coming from group algebras:

```
gap>L:=LieAlgebra(KG);;
gap>C:=LieCentre(L);
<Lie algebra of dimension 9 over GF(2)>
gap>D:=LieDerivedSubalgebra(L);
<Lie algebra of dimension 111 over GF(2)>
gap>Dimension(C)+Dimension(D)=Dimension(L);
true
```

We can again apply standard GAP functions such as for example

```
gap>CartanSubalgebra(L);;
<Lie algebra of dimension 30 over GF(5)>
```

12.2 Ordinary Characters

How to achieve a matrix representation for SL(2,5):

```
gap>SL25:=SL(2,5);;
gap>Irr(SL25);;
gap>IrreducibleRepresentationsDixon(SL25,last[2]);;
gap>Group(Image(last,GeneratorsOfGroup(SL25)));
Group([
 [ [ E(5)^4, -E(5)^2 ], [ E(5)+E(5)^3, -E(5)^4 ] ],
 [ [ 2*E(5)+E(5)^2+E(5)^3+E(5)^4, E(5)+E(5)^3 ], [ -E(5)-E(5)^2-E(5)^4, -E(5) ] ]
])
```

That's in characteristic 0 (the complex numbers ℂ or more concisely a cyclotomic field).

```
gap> DefaultField( [ E(5) ] );
CF(5)
gap> x:= Indeterminate( CF(5) );
x_1
gap> Factors( PolynomialRing( CF(5) ), x^5-1 );
[ x_1-1, x_1+(-E(5)), x_1+(-E(5)^2), x_1+(-E(5)^3), x_1+(-E(5)^4) ]
gap> f:=CF(5);
CF(5)
gap> b:=CanonicalBasis(f);
CanonicalBasis( CF(5) )
gap> Coefficients(b, E(5));
[ 1, 0, 0, 0 ]
gap> Coefficients(b, 1);
[ -1, -1, -1, -1 ]
gap> Coefficients(b, E(5)^2);
[ 0, 1, 0, 0 ]
gap> V:=FullRowSpace(f,2);
( CF(5)^2 )
gap> GeneratorsOfLeftModule(V);
[ [ 1, 0 ], [ 0, 1 ] ]
gap> coll:= [ [ 1, 1 ], [ E(5), E(5)^4 ], [ 1, E(5)^2 ], [ 1, E(5)^2 ], [ E(5), E ↩
    (5)^3 ] ];;
gap> V:= LeftModuleByGenerators( CF(5), coll );
<vector space over CF(5), with 5 generators>
```

12.3 Modular Characters

The needed representation can be achieved as follows:

```
gap>IrreducibleRepresentations(SL25,GF(5))[2];;
CompositionMapping(
[ (2,5,4,3)(6,11,16,21)(7,15,19,23)(8,12,20,24)(9,13,17,25)(10,14,18,22),
  (2,16,9)(3,21,15)(4,6,17)(5,11,23)(7,22,10)(8,12,13)(14,18,19)(20,24,25)
 ] -> [ [ [ Z(5)^3, 0*Z(5) ], [ Z(5)^0, Z(5) ] ],
  [ [ Z(5)^2, Z(5)^0 ], [ Z(5)^2, 0*Z(5) ] ] ], <action isomorphism> )
gap>Group(Image(last,GeneratorsOfGroup(SL25)));;
```

In the modular case one has to deal with Brauer tables and decomposition:

```
gap>ordtbl:=CharacterTable(SL25);;
gap>p:=5;;
gap>modtbl:=ordtbl mod p;
fail
gap>modtbl:=CharacterTable("2.A5") mod p;
BrauerTable( "2.A5", 5 )
gap>DecompositionMatrix(modtbl);;
gap>DecompositionMatrix(modtbl,1);
[[ 1, 0 ], [ 0, 1 ], [ 0, 1 ], [ 1, 1 ]]
gap>ordchars:=Irr(ordtbl){blocksinfo[1].ordchars};;
gap>rest:=RestrictedClassFunction(ordchars, modtbl);;
gap>modchars:=Irr(modtbl){blocksinfo[1].modchars};
[ Character( BrauerTable( "2.A5", 5 ), [ 1, 1, 1, 1, 1 ] ),
  Character( BrauerTable( "2.A5", 5 ), [ 3, 3, -1, 0, 0 ] ) ]
gap>dec:= Decomposition( modchars, rest, "nonnegative" );;
```

Furthermore there are integral representations. A (2,3)-generation can be looked at as expressing the group as a subgroup of PSL(2, $\mathbb{Z}$). See [56], [86] and [55] for more information.

Chapter 13

Lie Groups and Algebras

A hexagon looks similar to the root vectors of the A2 Lie algebra SU(3), whereas squares have some connection to the Lie Algebra B2 = C2, and the lattice D2 corresponds to a checkerboard and could serve as a model for a bishop world in Chess walking only on faces of one color.

We leave the computer for the moment and turn to some theory background.

The standard definitions for a Lie algebra are:

- a vector space is a free left module over a division ring R
- an algebra A is a vector space equipped with a bilinear map [cx+y,z] = c[x,z] + [y,z] $\forall \mathrm{x},\mathrm{y},\mathrm{z} \in A$, $c \in R$ (and same in second component)
- a Lie algebra L is an algebra with [x,x] = 0 (for R not of characteristic 2, anticommutativity [x,y] = -[y,x] is enough) and Jacobi identity: [x,[y,z]] + [y,[z,x]] + [z,[x,y]] = 0 $\forall \mathrm{x},\mathrm{y},\mathrm{z} \in L$

Any Lie algebra is a semidirect product of its radical and a semisimple subalgebra (Levi-Malcev decomposition).

13.1 General Theory

Lie groups are topological groups with a specified smooth structure.

See [7] to every Lie group G, we can associate a Lie algebra $\mathfrak{g}$ which completely captures the local structure of the group, at least if the Lie group is connected. A vector field associates a vector to every point in space, and a vector field on a Lie group G is said to be left-invariant if it commutes with left translation, which means the following. Define Lg[f](x)= f(gx) for any analytic function f : $G \mapsto$ some field F and all g, x in G. Then the vector field X is left-invariant if X Lg = Lg X for all g in G. The set of all vector fields on an analytic manifold is a Lie algebra over F. On a Lie group, the left-invariant vector fields form a subalgebra, the (finite-dimensional) Lie algebra associated with G, usually denoted by a gothic $\mathfrak{g}$. The association $G \mapsto \mathfrak{g}$ is a functor.

The Lie algebra has a basis of invariant vector fields that is taken by exponentiation into the space of left-invariant 1-forms on the Lie group. Such left-invariant 1-forms are called the Maurer-Cartan forms.

Every element v of the tangent space $\mathrm{T_e}$ at the identity element e of G determines a unique left-invariant vector field; the vector space underlying g may therefore be identified with $\mathrm{T_e}$. The Lie algebra structure on $\mathrm{T_e}$ can also be described as follows : the commutator operation $(\mathrm{x},\mathrm{y}) \mapsto \mathrm{x\ y\ x\hat{}-1\ y\hat{}-1}$ on G sends (e,e) to e, so its derivative yields a bilinear operation on $\mathrm{T_e}$ and satisfies the axioms of a Lie bracket.

Let's take a module V, then a representation of a group G is a map of groups $G \mapsto \mathrm{GL(V)}$ that is a homomorphism of groups. For a unitary representation, the map is to a unitary group U(n).

May $G \mapsto \mathrm{GL(V)}$ be a representation, then it's differential $\mathfrak{g} \mapsto \mathrm{End(V)}$ is a Lie algebra representation.

The homomorphism ad : $\mathfrak{g} \mapsto \mathrm{End(g)}$ is called the adjoint map, defined as:

$$\mathrm{ad(x)y} = [\mathrm{x},\mathrm{y}]$$

It's the differential of Ad, and commutes with the Lie bracket, so:

$$\mathrm{ad}[\mathrm{A},\mathrm{B}] = [\mathrm{ad\ A},\mathrm{ad\ B}]$$

In other words, the map sending each element to its adjoint action is a Lie algebra homomorphism of the original algebra into the Lie algebra of its derivations. In fact it's just a reformulation of the Jacobi identity. Another equivalent identity is the assertion that the action of any element on the algebra is a derivation; that is to say, it satisfies Leibniz's law, so we also have:

$$\mathrm{ad(x)[y,z] = [ad(x)y,z] \; + \; [y,ad(x)z]}$$

Taking the conjugation map of a group to itself, and then taking the differential of that, we get a very specific representation of a group, i.e. the adjoint represention of a group on its Lie algebra.

A Lie algebra is a logarithm of a Lie group, and a Lie group is an exponential of a Lie algebra. In the case of a compact connected matrix group, the exponential is even surjective and allows a parametrisation of such a group. The three notions of exp (complex analysis, Lie groups, and Riemannian geometry) are all linked together, the strongest link being between the Lie groups and Riemannian geometry definition. If G is a compact Lie group, it admits a left and right invariant Riemannian metric. With respect to that metric, the two exp maps agree on their common domain. In other words, one-parameter subgroups are geodesics.

The matrix-valued (formal power) series for the (analytical) functions are:

$$e^A = \sum_{n \geq 0} \frac{1}{n!} A^n$$

$$\ln(1+A) = \sum_{n \geq 1} -1^{n-1} A^n$$

The exponential series converges everywhere, and the series for the logarithm converges in a small enough neighbourhood of the origin.

As a consequence of the Baker-Campbell-Hausdorff formula:

Equation 13.1 Baker-Campbell-Hausdorff formula

$$e^A e^B = e^{A+B+1/2[\mathrm{A,B}]+1/12([\mathrm{A,[A,B]}]+[\mathrm{B,[B,A]}])+1/48([\mathrm{A,[B,[B,A]]}] - [\mathrm{B,[A,[A,B)}])))+...}$$

the local structure of a Lie group G near the identity (i.e. the rule for the product of two elements sufficiently close to the identity) is determined by its Lie algebra $\mathfrak{g}$.

The projective groups and their covers have the same algebra (let us be exact and say isomorph algebras instead) associated to them. And in the same manner non-isomorph algebras can have the same unique universal enveloping, associative algebra. The Lie algebra only determines uniquely the simply connected (universal) cover of a Lie group G.

13.2 Universal enveloping algebra

Universal enveloping algebras are the Lie theoretic analogues of group algebras. It is the algebra of left-invariant differential operators on G of all orders, with product the composition of differential operators.

The Lie bracket is not associative, but from any associative algebra we can cook up a Lie algebra by defining:

```
[X,Y] = XY - YX
```

and the universal enveloping algebra is the reverse process where for any Lie algebra we cook up an associative algebra such that:

```
XY-YX = [X,Y]
```

Let L be a finite dimensional Lie algebra with universal enveloping algebra U(L). Every (irreducible) L-module M is also an (irreducible) module for the enveloping algebra U(L). Over an algebraically closed field, then the center Z(L) of U(L) operates on every finite dimensional irreducible L-module via scalars. In the classical context, these scalars separate the irreducible modules, so that each finite dimensional irreducible module is uniquely determined by its central character. This fact depends on Harish-Chandra's Theorem, which relates Z(L) to polynomial invariants of the Weyl group of L.

In case of fields of positive characteristic (the modular case), the situation changes, among others the center of the enveloping algebra U(L) tends to be bigger. Given an n-dimensional Lie algebra L one can always find a subalgebra O(L) of Z(L) such that O(L) is a polynomial ring in n variables, and U(L) is a free O(L)-module of finite rank. One consequence is the finite dimensionality of irreducible L-modules, which contrasts with the classical situation. For algebraically closed fields the irreducible L-modules may thus again be subdivided according to their characters on O(L). Now different irreducibles may give rise to the same character. In fact, the irreducible modules belonging to the same character are modules over a finite dimensional associative algebra, the so-called reduced enveloping algebra. One obtains an algebraic family of finite dimensional algebras that is parametrized by the maximal spectrum of O(L).

13.3 Lie theory in Physics

The affine group is a semidirect group of GL (rotations) and the group of translations (the latter is the normal subgroup).

The Minkowski space is four dimensional space $\mathbb{R}^4$ with metric of signature (-1, 1, 1, 1). The Poincaré group (the inhomogeneous Lorentz group) is the group of isometries of the Minkowski space, that's to say all A with A$^+$gA = g whereas A$^+$ denotes the adjoint operator of A and g is a non-degenerate (det != 0), semidefinite (positive and negative Eigenvalues) metric like

```
gap>metric:=[
    [-1 0 0]
    [ 0 1 0]
    [ 0 0 1]
]
```

with signature 1 (an invariant, it's the sum of Eigenvalues) in this example.

In standard (3+1)-dim physics it is a 10-dimensional Lie group. Its unitary irreducible representations on Hilbert space are indexed by mass (nonnegative number) and spin (integer or half integer), and are associated with particles in quantum mechanics.

See [101] the Heisenberg algebra is a Lie algebra, and the Heisenberg groups are actually examples for nilpotent Lie groups not expressable as matrix groups (but as quotients of the matrix groups of unipotent matrices SUT by the center). Both the quotient and the embedding have the same Lie algebra with the canonical commutation relation [p,q] = 1. Besides (see [7]) there seems another naming convention to exist though identitfying the Heisenberg groups directly with some matrix groups and not some quotient.

To each physical observable corresponds a self-adjoint (Hermitian) operator O. In the Heisenberg picture the time-dependent observables satisfy (omitting explicit time-dependency)

$$\frac{dO}{dt} = [\,\mathrm{H}\,,\,\mathrm{O}\,]$$

whereby H is the Hamiltonian and [,] is the commutator. The Heisenberg picture is the formulation of matrix mechanics in an arbitrary basis, where the Hamiltonian is not necessarily diagonal.

The symmetries of a quantum system is the group of automorphisms of the algebra of operators. If a physical system has a symmetry group G, there is a unitary representation Π of G. This means that we get a unitary operator Π(g in G) satisfying Π(g_1 g_2) = Π(g_1)Π(g_2) ,i.e. the map Π from group elements to unitary operators is a homomorphism. The Π(g) act on O by taking an operator O to its conjugate $\Pi(\mathrm{g})O\Pi(\mathrm{g})^{-1}$. When G is a Lie group with Lie algebra $\mathfrak{g}$, differentiating Π gives a unitary representation π of $\mathfrak{g}$, and this map taking Lie algebra elements (with the Lie bracket in $\mathfrak{g}$) to skew-Hermitian operators (with commutator of operators) is a homomorphism. On O, $\mathfrak{g}$ acts by the differential of the conjugation action of G, this action is just that of taking the commutator.

A state space of a quantum system with symmetry group G carries not only a unitary representation of G, but also a unitary representation of its Lie algebra $\mathfrak{g}$, or equivalently, an action of the algebra U($\mathfrak{g}$). In this way a representation π gives a sub-algebra of the algebra O of observables.

The maximal Abelian subalgebra is about all observables to be measured simultaneously. The Casimirs are the generators of the center of the universal enveloping algebra.

13.4 SL(2,5) and sl(2,5)

Let's use SL(2,5) as a running example:

We achieved matrices of two generators in Chapter 12:

```
gap>SL25:=SL(IsMatrixGroup,2,5);;
gap>GeneratorsOfGroup(SL25);
[ [[ Z(5)  , 0*Z(5) ], [ 0*Z(5), Z(5)^3 ]],
  [[ Z(5)^2, Z(5)^0 ], [ Z(5)^2, 0*Z(5) ]] ]
```

Three 2-dim generators (with determinant 1) over GF(5) are . . .
And the corresponding Lie algebra sl(2,5) has as well 3 generators h,x,y (with trace zero):

```
gap>mat:=[ [[Z(5)^0,0*Z(5)],[0*Z(5),-Z(5)^0]],
           [[0*Z(5),Z(5)^0],[0*Z(5),0*Z(5)]],
           [[0*Z(5),0*Z(5)],[Z(5)^0,0*Z(5)]]];;
gap>sl25:=LieAlgebra(GF(5), mat);
<Lie algebra over GF(5)m with 3 generators>
gap>Size(sl25);
125
gap>SemiSimpleType(sl25);
"A1"
```

with [h,x]=2x, [h,y]=-2y and [x,y]=h. How do we construct them, starting with the matrix group?
We play around a little bit more:

```
gap>A:=FullMatrixAlgebra(GF(5),2);;
gap>L:=LieAlgebra(A);;
gap>L=FullMatrixLieAlgebra(GF(5),2);
true
gap>Dimension(L);
4
gap>Size(L);
625
gap>d:=DirectSumDecomposition(L);;
gap>d[2]=sl25;
true
gap>IsLieAbelian(sl25);
false
gap>IsLieNilpotent(sl25);
false
gap>IsLieSolvable(sl25);
false
gap>CartanSubalgebra(sl25);
<Lie algebra of dimension 1 over GF(5)>
gap>R:=RootSystem(sl25);
<root system of rank 1>
gap>PositiveRoots(R);
[[ Z(5) ]];
gap>PositiveRootVectors(R);
[ LieObject( [[ 0*Z(5), 0*Z(5) ], [Z(5)^0, 0*Z(5) ]] ) ]
gap>NegativeRoots(R);
[[ Z(5)^3 ]]
gap>SimpleSystem(R);
[[ Z(5) ]]
gap>C:=CartanMatrix(R);
[[ 2 ]]
gap>BilinearFormMat(R);
[[ Z(5)^3 ]]
gap>W:=WeylGroup(R);
Group([[[ -1 ]]])
gap>IsRestrictedLieAlgebra(sl25);
true
gap>B:=Basis(sl25);;
gap>SCT:=StructureConstantsTable(B);;
gap>l:=LieAlgebraByStructureConstants(GF(5),SCT);;
gap>ll:=IsomorphismSCAlgebra(B);;
gap>KillingMatrix(B);
[[ Z(5)^3, 0*Z(5), 0*Z(5) ], [ 0*Z(5), 0*Z(5), Z(5)^2 ],
 [0*Z(5), Z(5)^2, 0*Z(5) ]]
gap>BB:=Basis(L);;
gap>IsNilpotentElement(L, BB[2]);
true
gap>sl2:=FindSl2(L, BB[2]);;
gap>sl2=sl25;
true
gap>IsPerfect(sl25);
true
gap>sl25=DerivedSubgroup(sl25);
true
gap>u:=UniversalEnvelopingAlgebra(sl25,B);; Dimension(u);
infinity
```

Cayley-Hamilton and Jordan normalforms are about essentially reducing the problem of computing the exponential of a matrix to dealing with the exponential function over the underlying finite field.

```
gap>JordanDecomposition(SL25);
[[[ Z(5), 0*Z(5) ], [ 0*Z(5), Z(5)^3 ]],
 [[ 0*Z(5), 0*Z(5) ], [ 0*Z(5), 0*Z(5) ]]]
```

13.5 McKay Correspondence

The standard Platonic polyhedra correspond to the exceptional Lie algebras E6, E7, and E8 (see [100] and [109]).

Chapter 14

Algebra

14.1 Commutative Algebra and Algebraic Geometry

The symmetry group of a polyhedron is normally seen as s.th. composed of bulk rotations (sometimes together with reflections as well). The tetrahedron gets $\mathrm{A}(4) \cong \mathrm{PSL}(2,3)$, the cube $\mathrm{S}(4) \cong \mathrm{PGL}(2,3)$ (well, the Schur cover of S(4) is not unique though), and the icosahedron $\mathrm{A}(5) \cong \mathrm{PSL}(2,4) \cong \mathrm{PSL}(2,5)$.

The icosahedral group H(3) is the set of all rigid motions preserving an icosahedron. It is isomorphic to the direct product of the alternating group A(5) and the two-element group C(2). The subgroup isomorphic to A(5) acts as a set of rotations, and is denoted by H(3,+) (so reflections are left out now). If the icosahedron is situated so that the center of the icosahedron and the origin coincide, then the subgroup H(3,+) occurs as a subgroup of the orthogonal group SO(3), the rotations in 3-space fixing the origin.

The automorphism group (or symmetry group) of a lattice is the set of distance-preserving transformations (or isometries) of the space that fix the origin and take the lattice to itself (the hexagonal lattice gets a dihedral group of order 12), and furthermore adjoining the translations in lattice vectors gives the affine automorphisms (infinite group).

The (positive definite) quadratic form by which one can compare the vectors' length even in case they are not parallel / antiparallel is needed. One quadratic form for the hexagonal lattice is $\xi_1^2 + \xi_1\xi_2 + \xi_2^2$ and is (bi-)invariant under the symmetry group. But such forms may hide underlying symmetry, so a coordinate-free approach is preferred.

```
gap>GG:=SL(IsMatrixGroup,2,5);;
gap>m:=InvariantBilinearForm(GG).matrix;
[[ 0*Z(5), Z(5)^0 ], [ Z(5)^2, 0*Z(5) ]]
```

Besides there are polyhedral equations for the invariants retrieved using tools such as stereographic projection (it's of course quite ugly to involve extrinsic data from ambient space IMHO), Hessian and Jacobian forms. For example a tetrahedron gets $\Phi\hat{}3 - 12\sqrt{3}\,\mathrm{i}\,\Omega\hat{}2 - \Psi\hat{}3 = 0$, the cube gets $108\,\Omega\hat{}4 + ((\Phi\hat{}3 + \Psi\hat{}3)/2)\hat{}2 - (\Phi\Psi)\hat{}3 = 0$, and the icosahedron gets $1728\,\mathrm{I}\hat{}5 - \mathrm{J}\hat{}2 - \mathrm{H}\hat{}3 = 0$.

The latter is then used to solve the general polynomial of 5. degree involving Tschirnhausen transformation and hypergeometric functions. In general there are connections to modular functions. So obviously the question is if our generalized polyheda can be a starting point to attack higher polynomials. By the way those things get involved as well in case of iterative solutions (such as Newton's method), and the procedure then is only guaranteed to succeed if the Galois group is appropriate (see [90]).

The resulting (embedded) curves from the pieces movings in Section 3.3 look aesthetical and smooth.

Now here is some algebraic curves homework for you: take Klein's quartic $\mathrm{x}\hat{}3\mathrm{y} + \mathrm{y}\hat{}3\mathrm{z} + \mathrm{z}\hat{}3\mathrm{x} = 0$ (in homogeneous coordinates), remember the L2(7) group, and then play with the GAP curves package, compute the Riemann-Roch space of Klein's quartic over GF(7) for some divisors. Finally generalize all this for our framework:-)

If G is a finite subgroup of SL(2,C) or SO(3), there exists a complex reflection group 'G' containing G with ['G':G] = 2 (so both the polyhedral groups and their (binary) covers can be constructed this way).

14.2 Ideal Theory, CoInvariant Graded Algebras

A ring (resp. module) is called Noetherian if every ascending chain of ideals (submodules) becomes eventually stationary. According to the Hilbert's Basis Theorem every finitely generated module over a finitely generated K-algebra is Noetherian (see [97], [96] or others). Please consider to think about Hilbert's (projective) Nullstellensatz

Equation 14.1 Hilbert's Nullstellensatz

$$I(V(I)) = \sqrt{I}$$

Here the definition of the radical ideal differs from the one in Section 12.1.

Due to Noether and moreover since finite groups are (geometrically) reductive, their invariant rings are finitely generated and graded. The Hilbert-Poincaré series encodes the dimensions of the ideals of generators of given degree. A graded algebra is Cohen-Macaulay if and only if it is free as a module over a subalgebra generated by a homogeneous system of parameters. A polynomial ring is Cohen-Macaulay. For a finite group G, the invariant ring is polynomial iff (if and only if) G is a generalized reflection group.

The theory depends on what representation is taken. V denotes a faithful irreducible representation and K[V]^G the invariant ring of G in what follows. If char(K) does not divide the group order, then K[V]^G is Cohen-Macaulay. In the modular case the invariant ring is in general not that nice, and the Molien series gives only the so called extended Hilbert series $\hat{H}(\mathrm{K[V]}^G)(t)$.

```
gap>ctl:=CharacterTable(AlternatingGroup(5));
gap>psi:=First(Irr(ctl),x->Degree(x)=3);
Character( CharacterTable( Alt( [ 1 .. 5 ] ) ),
[ 3, -1, 0, -E(5)^2-E(5)^3, -E(5)-E(5)^4 ] )
gap>MolienSeries(psi);
( 1-z^2-z^3+z^6+z^7-z^9 ) / ( (1-z^5)*(1-z^3)*(1-z^2)^2 )
```

How about the inverse problem: Given some invariant ring R, construct a group such that R is the group's invariant ring.

14.3 Gröbner Bases, Buchberger's Algorithm, Syzygies

See [99], [94] and [98] for general information on algorithmic invariant theory. The Buchberger algorithm (involving syzygies) delivers Gröbner bases for fundamental systems of invariant rings.

A permutation group example:

```
gap>LoadPackage("singular");;
gap>G:=AlternatingGroup(5);;
gap>R:=PolynomialRing(GF(2),5);
GF(2)[x_1,x_2,x_3,x_4,x_5]
gap>GeneratorsOfInvariantRing(R,G);
[ x_1+x_2+x_3+x_4+x_5,
  x_1*x_2+x_1*x_3+x_1*x_4+x_1*x_5+x_2*x_3+x_2*x_4+x_2*x_5+x_3*x_4+x_3*x_5+x_4*x_5 ↩
      ,
  x_1*x_2*x_3+x_1*x_2*x_4+x_1*x_2*x_5+x_1*x_3*x_4+x_1*x_3*x_5+x_1*x_4*x_5+x_2*\
x_3*x_4+x_2*x_3*x_5+x_2*x_4*x_5+x_3*x_4*x_5,
  x_1*x_2*x_3*x_4+x_1*x_2*x_3*x_5+x_1*x_2*x_4*x_5+x_1*x_3*x_4*x_5+x_2*x_3*x_4*\
x_5, x_1*x_2*x_3*x_4*x_5 ]
gap>I:=Ideal(R,last);
<two-sided ideal in GF(2)[x_1,x_2,x_3,x_4,x_5],
  (5 generators)>
gap>GroebnerBasis(I);
[ x_1+x_2+x_3+x_4+x_5,
  x_2^2+x_2*x_3+x_2*x_4+x_2*x_5+x_3^2+x_3*x_4+x_3*x_5+x_4^2+x_4*x_5+x_5^2,
  x_3^3+x_3^2*x_4+x_3^2*x_5+x_3*x_4^2+x_3*x_4*x_5+x_3*x_5^2+x_4^3+x_4^2*x_5+x_\
4*x_5^2+x_5^3, x_4^4+x_4^3*x_5+x_4^2*x_5^2+x_4*x_5^3+x_5^4, x_5^5 ]
```

The matrix representation:

```
gap>GG:=SL(IsMatrixGroup,2,5);;
gap>RR:=PolynomialRing(GF(5),2);
GF(5)[x_1,x_2]
gap>GeneratorsOfInvariantRing(RR,GG);
[ x_1^5*x_2-x_1*x_2^5,
  x_1^20+x_1^16*x_2^4+x_1^12*x_2^8+x_1^8*x_2^12+x_1^4*x_2^16+x_2^20 ]
gap>II:=Ideal(RR,last);
<two-sided ideal in GF(5)[x_1,x_2], (2 generators)>
gap>GroebnerBasis(II);
[ x_1^5*x_2-x_1*x_2^5, x_1^20-x_1^4*x_2^16+x_2^20, x_2^21 ]
gap>ReducedGroebnerBasis(II,MonomialLexOrdering());
[ x_2^21, x_1^5*x_2-x_1*x_2^5, x_1^20-x_1^4*x_2^16+x_2^20 ]
```

Using the SINGULAR system directly:

```
>LIB "finvar.lib";
>ring R=5,(x,y),dp;
>matrix A[2][2]=0,4,1,0;
>matrix B[2][2]=0,1,4,4;
>matrix P,S,IS=invariant_ring(A,B);
      //returns primary invariants, and second secondary invariants, i.e. module  ↩
          generators
      //over a Noetherian normalization, and irreducible secondary invariants
      //if the Molien series was available
>print(P);
x5y-xy5
>print(S);
x20+x16y4+x12y8+x8y12+x4y16+y20
>print(IS);
1
>ideal I=P,S;
>ideal J=std(I);
>J;
J[1]=x5y-xy5
J[2]=x20-x4y16+y20
J[3]=y21
```

One speaks of primary invariants $f_1, ..., f_n \in \mathrm{K}[\mathrm{V}]^G$ and homogeneous, secondary invariants $\mathrm{g}_1, ..., \mathrm{g}_\mathrm{n}$ such that $\mathrm{K}[\mathrm{V}]^G$ is generated as a module over $\mathrm{K}[\mathrm{f}_1, ..., \mathrm{f}_\mathrm{n}]$. They are not uniquely determined, and being a primary or secondary invariant is not an intrinsic property of an invariant. Fundamental invariants are a minimal system of generators of $\mathrm{K}[\mathrm{V}]^G$.

Syzygies are algebraic relations between fundamental invariants.

```
>setring R;
>module M=syz(I);
>M;
M[1]=x20*gen(1)+x16y4*gen(1)+x12y8*gen(1)+x8y12*gen(1)+x4y16*gen(1)+y20*gen(1)
     -x5y*gen(2)+xy5*gen(2)
```

Let's use some more methods from the 'finvar' library:

```
>list L=primary_invariants(A,B);
>L;
[1]:
   _[1,1]=x5y-xy5
   _[1,2]=x20+x16y4+x12y8+x8y12+x4y16+y20
>matrix SE=secondary_not_cohen_macaulay(L[1],A,B);
>print(SE);
1
```

The Gröbner bases are confluent and enable you to decide ideal membership.

```
>poly f=x10y10;
>reduce(f,J,1);       //3rd parameter 1 avoids tail reduction
x2y18                 //f is not in I
>poly g=y2*J[1]-2x*J[2];
>reduce(g,J,1);
```

```
0                    //g is in I
>matrix N=lift(J,g);
>N;
N[1,1]=y2;
N[2,1]=-2x;
N[3,1]=0
```

A minimal free resolution of the finitely generated R-module Coker(matrix(J)) = F_0 / J with finitely generated free R-modules F_i $\forall i \geq 0$ is an exact sequence

$$\cdots \to F_k \to \cdots \to F_1 \to F_0 \to \frac{F_0}{J} \to 0$$

with finite minimal length n if $\forall k, k > n$: $F_k = 0$. The rank(F_k) $\forall k \geq 0$ are called the k-th Betti number of the module F_0 / J.

```
>resolution Re=mres(prune(J),0);
>Re;
 1    2    1
R <--R <--R

0    1    2
>print(Re);
[1]:
   _[1]=x5y-xy5
   _[2]=x20-x4y16+y20
[2]:
   _[1]=x20*gen(1)-x4y16*gen(1)+y20*gen(1)-x5y*gen(2)+xy5*gen(2)
>print(betti(Re), "betti");
...
total:     1     2     1
```

Hilbert's Syzygy theorem tells us that for any monomial ordering on K[x] = K[x_1,...,x_n] and R the associated ring, any finitely generated R-module M has a free resolution of length $\leq n$ (the number of variables).

The annihilator of a module M = Coker(B) given by a presentation matrix B over the quotient ring R / I is (by definition) the ideal $< 0 > :$ M .

```
>qring Q=groebner(I);    //defines the quotient ring Q = R/I
                         //the command groebner() uses (in contrast to std()) a
      //Hilbert series based standard basis approach (if available)
>Q;
//   characteristic : 5
//   number of vars : 2
//        block   1 : ordering dp
//                  : names    x y
//        block   2 : ordering C
// quotient ring from ideal
_[1]=x5y-xy5
_[2]=x20-x4y16+y20
_[3]=y21
>module B=[y2],[-2x];
>ideal ann=quotient(B,freemodule(nrows(B)));
>ann;
ann[1]=x
ann[2]=y2
```

For a Noetherian local ring A and a finitely generated A-module M, we always have depth(M) $\leq$ dim(M)

Modules with depth(M) = dim(M) are called Cohen-Macaulay modules, A is called a Cohen-Macaulay ring if it is a Cohen-Macaulay A-module. Regular local rings are Cohen-Macaulay.

Finally have a look at the HAP package (see [66]).

Chapter 15

Lattices

15.1 Icosians

2.A(5) can be expressed over the quaternions as a finite group of order 120 (it's a subgroup of Sp(1), the 1x1 quaternionic symplectic group) and is isomorph to SL(2,5). The Sylow 2-subgroup of 2.A(5) is Q_8.

A perfect group is isomorph to it's derived subgroup. SL(2,5) is said to be the only finite perfect group admitting a fixed-point-free representation, and it's the fundamental group of the Poincaré homology sphere (the only known finite fundamental group of such a homology sphere).

The irreducible representations of the Binary Icosahedral Group are intimately related to the exceptional root lattice E(8), the lowest-rank even unimodular (that is det = 1) lattice in existence.

See [89] the icosian ring is the set of all finite sums of elements in the icosian group. Elements of this ring are simply called icosians. A typical icosian has the quaternionic form $\alpha + \beta i + \gamma j + \delta k$ where the coordinates belong to the golden field Q(τ) with $\tau = \frac{1+\sqrt{5}}{2}$ and so have the form a + b sqrt(5) with $a, b \in \mathbb{Q}$. The icosians of quaternionic norm 1 are the elements of the icosian group. With respect to the quaternionic norm the icosians belong to a 4-dim space over Q(τ) ; with the Euclidean norm they lie in an 8-dim space, then the icosian ring is isomorphic to an E(8) lattice.

```
gap>Indeterminate(Rationals,"x");;
gap>p:=UnivariatePolynomial(Rationals,[-5, 0, 1],1);;
gap>e:=FieldExtension(Rationals,p);;
gap>Q:=QuaternionAlgebra(e);
gap>b:=BasisVectors(Basis(Q));
gap>NormQuat := function( quat )
        if not IsQuaternion( quat ) then
          Error( "<quat> must be a quaternion" );
        fi;
        return Sum( List( ExtRepOfObj( quat ), c -> c^2 ) );
    end;
```

Another aspect is the epimorphism from 2.A(5) to A(5) which can be taken from [89].

15.2 Designs, Codes

The (5-transitive) sporadic Mathieu group M(24) arises as the automorphism group of the Steiner system S(5,8,24). Another construction involves PSL(2,23) (it is one of the maximal subgroups) and the Golay code. This translates to a 11-dim linear representation over GF(2) (see [48]):

```
gap>gen1 := [
  [0,1,0,0,0,0,0,0,0,0,0],
  [1,0,0,0,0,0,0,0,0,0,0],
  [0,0,0,1,0,0,0,0,0,0,0],
  [0,0,1,0,0,0,0,0,0,0,0],
  [0,0,0,0,0,1,0,0,0,0,0],
  [0,0,0,0,1,0,0,0,0,0,0],
  [0,0,0,0,0,0,0,1,0,0,0],
  [0,0,0,0,0,0,1,0,0,0,0],
  [0,0,0,0,0,0,0,0,0,1,0],
  [0,0,0,0,0,0,0,0,1,0,0],
  [0,0,1,1,0,0,0,0,1,1,1]
  ]*Z(2);
gap>gen2 := [
  [0,0,1,0,0,0,0,0,0,0,0],
  [0,1,1,0,0,0,0,0,0,0,0],
  [1,0,1,0,0,0,0,0,0,0,0],
  [0,0,0,0,1,0,0,0,0,0,0],
  [0,0,0,0,0,0,1,0,0,0,0],
  [1,0,1,1,1,1,0,0,0,0,0],
  [0,0,0,1,0,0,0,0,0,0,0],
  [0,0,0,0,0,0,0,0,1,0,0],
  [0,0,0,0,0,0,0,0,0,0,1],
  [1,0,0,0,0,0,0,0,0,1,0],
  [0,0,0,0,0,0,0,1,0,0,0]
  ]*Z(2);
gap>M:=GModuleByMats([gen1,gen2],GF(2));;
gap>MTX.IsAbsolutelyIrreducible(M);
true
gap>MTX.CompositionFactors(M);
gap>G:=Group(gen1,gen2);;
gap>DisplayCompositionSeries(G);
G(size 244823040)
 | M(24)
1 (size 1)
```

See [89] there are various further topics such as constructions of lattices from integral representations, of codes from permutation representations, and of spherical codes and designs from orthogonal representations.

15.3 Packings, Coverings and Combinatorics

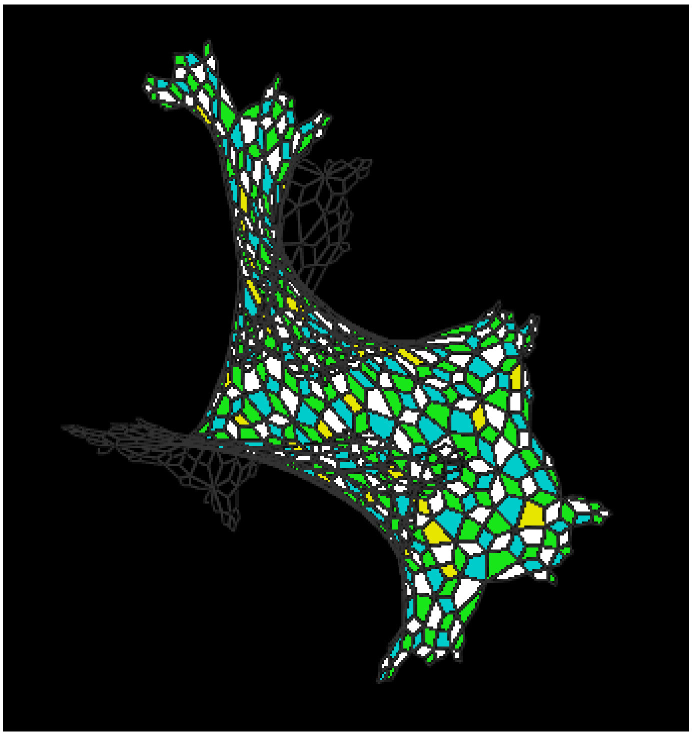
Non-lattice packings, Latin magic squares (connection with knight's tour and coloring) could be looked at.

(See [169]) analog to Chess' 8 - queens problem (take type B pieces and others) one can discover interesting combinatorics (perhaps using backtracking).

A related problem is the hard hexagon entropy constant (turns out to be algebraic), defined as the limes of N-th root of the number of configurations of nonattacking kings on a n × n Chessboard with regular hexagonal cells, where $N \equiv n^2$. And the cousin to this non-attacking, maximal packing problem is the minimal covering problem asking how many knights (queens, kings ...) it takes to control all fields of the board.

15.4 Quasicrystals

Those graphs are repetitive, but aperiodic (no translation symmetry). In Figure 15.1 you can see (an intermediate state of the 3D layouting of) how a decagon has developed under four inflation/deflation iterations (see for example [130] and Section 7.31 for more pictures (not relying on any fixed angles).



a Penrose quasicrystal (after applying some inflation/deflation rule in 4.th. iteration) on its way to an optimised layout

Figure 15.1: *Penrose Quasicrystal in Progress*

The Gaussian curvature is intrinsic and invariant under isometries, and a good embedding into ambient space should not introduce more (mean) curvature than necessary.

The boundary looks self-similar.

One could glue some edges together (as in Section 7.30) to make full use of 3D, besides thus one could achieve quasicrystals without a boundary.

15.5 Modular functions

There is an intimate relation to number theory, for example using the triangular planar tiling with $\gamma = \mathbb{Z}[\xi_3]$ the ring of Eisenstein integers what we quickly describe now. Let $\mathrm{l}_2(\mathrm{m})$ denote the number of sublattices with index m, then the Dirichlet series generating function F_2 can be expressed using Riemann's Zeta function:

Equation 15.1 Dirichlet Series Generating Function F2

$F_2(s) = \sum_{m=1}^{\infty} \frac{l_2(m)}{m^s} = \zeta(s)\zeta(\mathrm{s}-1)$

15.6 ToDo

- The number of boundary edges goes (10, 20, 50, 120, 290, ...) and seems to follow the rule 2*pre + prepre; the number of faces increases as (10, 30, 90, 260, 740, ...), this integer sequence is obeying which rule?

- Theta series (how many faces are in the shell at distance $n \in \mathbb{N}$ for a given type of piece and graph)

Chapter 16

Geometry

16.1 Riemannian Surfaces

The three polyhedral examples, L_27 etc. arise as well in the context of Hurwitz inertia groups (automorphism groups of compact Riemannian surfaces, factor groups, torsion-free Fuchsian groups, Galois coverings, see [51]).

16.2 Difference Geometry and Metric Spaces

The torsion tensor depends on a connection. The abstract manifolds department puts it this way (here X and Y are some vector fields):

$$\tau = \nabla_X Y - \nabla_Y \mathrm{X} - [\mathrm{X}, \mathrm{Y}]$$

The curvature is defined as follows:

$$\kappa = \nabla_X \nabla_Y - \nabla_Y \nabla_X - \nabla_{[\mathrm{X},\mathrm{Y})}$$

Those equations have counterparts in tensor formulation and as well in the context of forms. In the latter case they are also called Cartan's first (relating the Lie derivative to the inner and exterior derivative) and second structure theorem.

The holonomy of a connection can be identified with a Lie group, the holonomy group. A subspace of its Lie algebra is closely related to the curvature of the connection (via the Ambrose-Singer theorem).

The global Gauss-Bonnet theorem expresses the Euler-Poincaré characteristic of a manifold as a curvature integral.

Chapter 17

Differential Games and Mathematical Physics

17.1 Differential Games

The following three laws of motion form the basis for classical mechanics:

- In the absence of force, a body either is at rest or moves in a straight line with constant speed.
- Force is proportional to the time derivative of momentum.
- Whenever a first body exerts a force on a second body, the second body exerts a force on the first body equal in magnitude and opposite in direction (actio et reactio).

— Sir. Isaac Newton

Perhaps the term body also applies to a Chess piece.

In Go things such as invading, sliding in the opponent's territory (by monkey jump etc.) seem to obey some potential field laws. Dealing with moyo and thickness, often the term 'radiating influence' us used. Temperature used in combinatoric game theory comes from thermodynamics, it's the (Eulerian) integrating denominator such that the resulting differential (the entropy) becomes exact. A game can burn slowly in a steady fashion, or it might develop hot fighting.

Due to the similarities one could try to apply the methods from mathematical physics to games.

17.2 Discrete Mechanics

Let's treat the following abstract 2-body problem, a system of two coupled linear difference equations, which can (since they are linear this is for sure) be rewritten to make them explicit in highest order and then be used to run a forward simulation (simultaneously or better taking turns). In Equation 17.1 we have fancied up s.th. (the bold font indicates vectors, and t denotes the time coordinate):

$$\frac{d^2\mathbf{r}_1}{dt^2} + (\mathbf{r}_1 - \mathbf{r}_2) = 0$$

$$\frac{d^2\mathbf{r}_2}{dt^2} + (\mathbf{r}_2 - \mathbf{r}_1) = 0$$

Equation 17.1 Simple Dynamical 2-Body System

(await next release for an animation)

This system is an integrable system and should be solved analytically (perhaps using z-transformations). The Noether theorem might be involved. Drawn positions correspond to periodic solutions.

The general equivalence theorem states that the physical laws have the same structure in all coordinate systems. Well, does that mean that there must be a Chess theory delivering the optimal play for all possible positions?

17.3 Quantum Mechanics

The classical buzzwords must appear here: particle-wave dualism, uncertainty principle, Heisenberg picture

17.4 Difference Equations and Cellular Automata

Mathematical Physics is based on differential equations. In the discrete case it is sometimes treated in the context of recursions. The Hamiltonian is an invariant and describes the evolution in time.

17.5 ToDo

- put on every edge a standard resistor of 1 Ohm, and then for every 2 given vertices, make use of Kirchhoff's laws and give an analytic solution about the resulting resistance (see [144] for a random walk approach, other references offer a straightforward linear algebra treatment).
- what sort of knots and links arise from the piece movements
- vibrational modes using character theory (assume the edges as physical springs, see [74])
- Percolation and Ising models

Chapter 18

Chess Pieces Groups

18.1 2-Body Systems

Once the single piece case is done, of course the real fun just begins. Chess is a differential game, meaning that the locations of the pieces at subsequent times are related.

18.1.1 King A versus King A

Let's show how a physical treatment would look like for two pieces of type A and opposite colour on a graph without boundary, for example a torus.

We need a group (and moreover one of minimal size) reaching all (drawn) positions and all legal moves between them, involving the least possible effort on our side in the construction process. There are 2 internal degrees of freedom, so imagine 4 new generators as follows:

- the ensemble of two pieces is walking on the 2-dim surface, that will take 2 generators
- the local interaction between the two pieces will involve behavior similar to let's say the Kepler problem, the radial and angular modes deliver another 2 generators

Or we better simply think of a relative movement in the two directions separately (and doing them alternatingly or in parallel delivers some sort of modes).

The generators don't commute, or do they?! Actio = Reactio is involved probably. After all we want a group, and that's why once one piece is reflected (it avoids being captured), the other mustn't continue as if nothing had happened. And again we'll have some flags indicating a reflected state, so the situation can be distinguished from that achieved by simple translation. Well, the modes are probably not what we will dump out directly as permutations, but (as in the single piece case) we work locally and assign labels using as much information as is necessary to be faithful. Actually (at the moment) we think that the _intended_ locations (of both pieces) at times t=0,1 are needed (same as for the case of a single piece), of course normalized to where the "a"s are pointing to. It's not necessarily the actual locations arising in due course, but (actually similar to the tricks (low-high flipping) used for the case when two boundary lines are neighboured) one ply before a central collision the piece not at turn still intends to go straight forward, this is important because when the piece at turn changes its mind and rotates (does "b" instead of "a"), this original direction will be realized. Additionally the reflection states are also taken into account, for each piece separately (whereby it doesn't make a difference whether the reflection takes place at a boundary or by interaction with another piece). Because we don't use single piece generators but only combined stuff like (a,a), we must set this doubledly (a one piece ply apart) to reach all situations in general (parity).

For the record we also investigated the approach to use higher orders (that is the final real locations at times t=0,1,2) for the labels, thereby obviously the evolution in time can be modelled as well, but putting the solutions into relation to each other (say this one is (a,a), then what is (b,a)) didn't work out. And of course just the _actual_ locations at t=0,1 aren't enough. since we can't be sure that we are in the middle of a collision when our labelling procedure is beginning, whereas we must distinguish between a half-reflected and a simple translation situation. We hope that now everything is clearly explained :-)

It turned out that it is a good strategy to simply ensure that the 'other' piece (that is the piece not having as the first one to change direction to avoid suicide, it only changes direction as a reactio effect) should bounce against a collision place when going the further development backwards in time. It is not necessarily the original collision place, but in case the two locations of the pieces are neighboured over multiple faces then one might decide to use another collision place for the bouncing, deciding on criteria like the conservation of angular momentum rather than aesthetic grounds like avoiding an abrupt back-reflection into itself whereby the latter nonetheless often serves as a good hint signalling the aforementioned problem.

And we applied some rotations (before the collision is taking place to the piece not to move next, or in other words to the other piece after the collision) in order to see if such a position would result in a bad intended move by the piece to move next (thus invoking a procedure similar to the bouncing treatment), because neighboured repairings make it necessary to do the same for other situations as well in order to avoid 'disjoint-and-duplicate-free' conflicts.

On zonotope_5 and deltoid (the odd poly clearly showed the bouncing idea necessity) we ended up with using two iterations to arrive at the final new directions for the two pieces, that is to say the high/low flip occured twice, since after the first exchange of momenta and subsequent tweaking around we still found the piece to move in a suicidal position. We don't know yet if this is only a curiosity, perhaps even unnecessarily blowing up our groups, or if there are subtle interaction rules behind it. First of all we were already satisfied with one working group for the moment fulfilling our purposes.

Now some general words of wisdom about our method. It's already implicit that the players move alternatingly, and that they do not risk a loss by bad play.

One can state that we can work locally (we need not look around globally in order to decide how the orbits should evolve). Probably one can formulate the procedure within a difference equation framework, and it should be possible to prove that by this artificial restriction to obey some additional rules one doesn't lose any solution. In contrast, it will enable us to follow an approach to solve the stuff systematically instead of searching around.

In general the generators (a,a), (b,a) and (a,b) do not generate the full group. Thereby (b,a) means first of all the piece with number 1 moves in 'b' direction, then the other does an 'a'. For example on the trapezohedron graph we also need (c,a), (a,c). And one might think that on a torus (b,b) could be the same as (b,a)*(a,b)*(a,a)$^{-1}$, but this is not the case (only for some labels). But nevertheless adding (b,b) to the generators doesn't make a difference for this particular group, in general we better use some more additional generators like (c,a), (b,b) and (a,c) (but perhaps the results below are still only subgroups). The general situation will be investigated when the time is ready.

We clearly see how to dump out the stuff for the simple cases and so we really did, in Table 18.1 are some preliminary results for a system of two Chess pieces of type A, king roles and opposite colour (only drawn positions are dealt with so far):

Well, of course this way to present the result is not very informative. One would like to see how the overall movement (2-dim) can be separated from the other structures (movement relative against each other (in 2 dimensions), and links). When changing the board, what changes, and what stays the same? The cases where we weren't able to figure out the degree (that is the number of labels) of the permutation representation indicate that there is a certain need for improvement of the current strategy, for example constructing and storing the orbits for every possible starting position again and again is of course a waste of space and time. Much work to be done.

Cautious readers pointed out that boards with boundaries could lead to even higher levels of insight. Guess what, it is already on our agenda:-)

When there are boundaries around, we anticipate the following issues. First of all there is the additional topic called stalemate, which simply means one party might decide to do a Null-move, which mathematically still fits perfectly into the framework of group theory.

It is still completely trivial to decide what positions are drawn or not. Imagine an exotical board with a corridor attached to an area with enough space for a piece to move around and feel free a little bit. Place one piece in the middle of the corridor, and the other into the wide area. Then (because whereas a pass in general is not considered a valid move, in case of a stalemate it is) the (trapped) piece might dare to move further down into the corridor, it doesn't depend on how far away from the entry the other piece is.

But another issue troubling us happens when the two pieces as an ensemble approach a boundary, let us say on a board with faces having four sides (poly=4). The first piece to move reaches the boundary and gets reflected, ready to move on in a reflected mood. But then the other piece follows up immediately, thus squeezing the unlucky guy. What shall he do? He might go to the left or to the right, probably depending on his (high/low) former reflection status, and the other piece will simply be reflected back. Ok, but now all labels are taken, and we have other situations arriving here. The first piece might have gotten trapped by another piece coming not from the back but from the right or left, leading to the same

graph	degree	group (lossy)
dodecahedron	2400	$Z(2)^{19} \times A(5)^{20} \times A(20)$
icosahedron	5040	$Z(2)^{41} \times A(5)^{42} \times A(42)$
deltoid	1440	$Z(2)^{179} \times Z(3)^{60} \times A(60)$
delt_trunc	828	
tetra_iter_stl_3_1	792	
tetra_stl_II	out of memory:-(	
doubletetrahedron	18	$Z(3) \times A(5)$
bipyramid_penta	360	$Z(2)^{35} \times Z(5)^{36} \times A(36)$
cube_triang_I (bipyramid_hexa)	648	$Z(2)^{53} \times Z(3)^{30} \times A(27)$
bipyramid_octa	1440	$Z(2)^{180} \times A(45)$
cube_triang_II	1332	
yabi	2052	
oct_stl_I	3024	
tetra_reticul	out of memory:-(	
oct_stl_II	out of memory:-(	
torus_3x4	216	$Z(2)^{19} \times Z(3)^{11}$
trapezohedron	192	$Z(2)^{34}$
torus_hexa_3_5	960	$Z(2)^{31} \times Z(3)^{32} \times Z(5)^{31} \times A(32)$
doublecube	512	$Z(2)^{126} \times A(32)$
triplecube	1536	$Z(2)^{376} \times A(96)$
ico_stl_I	out of memory:-(	
ico_stl_II	out of memory:-(	
doubledodeca	out of memory:-(	
L2_7	out of memory:-(	
L2_7_dual	out of memory:-(	
S5	10560	
A6	832	$Z(2)^{309} \times A(104)$
zonotope_5	4080	
rhombic_dodeca	960	$Z(2)^{79} \times Z(3)^{20} \times A(20)$
rubik_cube_2x2	6336	
icositetra	out of memory:-(	
dode_tor_6	out of memory:-(	
hexa_tor	out of memory:-(	

Table 18.1: *2-Body Groups*

result (the other possible further developments are taken already by other input positions). Perhaps one can say it has to do with the sort of crossing explicitely noticed there (as there are the unknotted, the left-over right and the right-over-left crossing). We think that we must triple these labels. This somehow resembles the trick we used in the single piece case when doubling the labels because of the reflection.

To cut a long story short, you will understand that the boundaries have to wait until the next release.

18.1.2 King C versus King A

To be done.

18.1.3 King F versus King A

Still to be done.

18.1.4 King B versus King A

Also to be done.

18.1.5 King C versus King C

Also to be done.

18.1.6 King F versus King C

To be done.

18.1.7 King B versus King C

To be done.

18.1.8 King F versus King F

Yes, don't forget this one.

18.1.9 King B versus King F

To be done.

18.1.10 King B versus King B

To be done.

18.2 3-Body 2-Party Systems

These systems will be even harder to treat, but we will manage.

18.2.1 King A, Std A versus King A

To be done.

18.3 General Remarks

The scenario from above is the periodical, drawn case. Stalemate is easily contained when some corners are present.

The idea is that the parties keep their strategy (chosen piece, direction and velocity) as long as they don't have to change their plans. It's a deterministical approach. We deal with local interactions of the Chess pieces, a king can only be captured by another piece in its neighbourhood (which also applies to long-ranging pieces).

The situation of two pieces of type A is trivial from a Chess player's perspective, and so the underlying mathematics should turn out to be feasable as well. When the board is increased, a naive alpha-beta search becomes unhandable, since it only stops at terminal nodes. The permutation representation approach will not be the final tool as well, because the number of labels grows too much, whereas the actual situation does not become more complicated. But in the dual space the calculations are scaling well. We have a general overall movement, an angular momentum, a relative movement, and some relative angular thing going on. And that's it.

We have got the impression that certain Brommann algebras are involved here (they still have to be written down to be accessible though).

> [. . .] It is hard to imagine a more spectacular application of representation theory [. . .]
>
> — [74]

The system of two interacting Chess pieces inherits the symmetry of the board's faces, i.e. the dihedral group $D_{2\text{poly}}$. The module in question can thus be expressed as a direct sum of irreducible $\mathbb{Z}D_{2\text{poly}}$ -modules. Therefore one only needs to calculate the character of the module and then express the module as a direct sum of homogeneous components. One finds exactly the modes corresponding to translation, rotation, and vibration (expand-contract and counterwise change of angle) which were implicitely used in the construction of the 2-body groups above. Speaking of reaching something like a first milestone, a correct local formulation of the Hamiltonian of the interaction could probably be achieved following this line of thought.

Chapter 19

Complexity Theory

19.1 NP != P and beyond

This part is about considerations how the effort of deterministic algorithms (for example depth-first searches) as well as probabilistic methods increases depending on the problem input size.

On the other hand one could focus on algorithms that perform well in practice rather than on those with the best theoretical complexity. But with advances in hardware, in the long run implementations with more favourable complexity also run faster in practical computation.

19.1.1 Introduction

(See [7]) the class P consists of all those decision problems that can be solved on a deterministic sequential machine in an amount of time that is polynomial in the size of the input; the class NP consists of all those decision problems whose positive solutions can be verified in polynomial time given the right information, or equivalently, whose solution can be found in polynomial time on a non-deterministic machine. NP-hard problems are those to which any problem in NP can be reduced in polynomial time. NP-complete problems are those NP-hard problems which are in NP. A number of problems which operate not on normal input but on a computational description of the input, are known to be EXPTIME-complete. Because it can be shown that $P \subsetneq$ EXPTIME , these problems are outside P, and so require more than polynomial time.

PSPACE = NPSPACE, but up to now it's not known whether P and NP, NP and PSPACE, PSPACE and EXPTIME, or EXPTIME and NEXPTIME are equal or not.

As already mentioned, general graph layouting and coloring are both NP-complete. And also the existence of a Hamiltonian cycle in a graph belongs to this class.

There are complexity considerations for typical tasks in group theory (see Chapter 11) as well. Things like computing the center, composition factors, Sylow subgroups of permutation groups are polynomial time, whereas finding isomorphisms of permutation groups or finding generators of automorphism groups of a graph fall into the so-called graph-isomorphism class. There exists no known P algorithm for graph-isomorphism testing, although the problem has also not been shown to be NP-complete, so some people invented a complexity class called 'graph isomorphism-complete' which is thought to be entirely disjoint from both NP-complete and from P. However, a polynomial time algorithm is known for planar graphs, graphs with restricted genus, graphs with bounded (maximal) degree of the vertices (valence) by a constant, or graphs with bounded Eigenvalue multiplicity (see for example [4], [14] (btw. using substantially computational group theory), [84] and [19]).

Computing Gröbner bases is EXPSPACE-complete (in general). So probably we won't have an easy going there, but in any case they will provide a nice way to organize results once they have been achieved (and in our geometrical setting certainly the complexity is not similar to the worst case estimation anyway).

19.1.2 Games

Now dealing with the strategy games, extending the actually finite problems to so called n × n board games (increasing n could translate into stellation (finer grained meshgrids). A general journey through the references tells us that some variants of n-in-a-row like games might have a polynomial-time strategy solution this way or the other, Hi-Q (see [148]) and some Dots-and-Boxes problems (see [173]) are NP-complete or NP-hard, Reversi and Hex are in PSPACE-complete, Go (with Kos) using Japanese rules as well as Checkers and Chess are in EXPTIME-complete (see also [7]).

Whilst we have just come to know that here all general hope could be lost, nevertheless there are well-defined strategies about how to win elementary endgames (KQ:K, KR:K, KQ:KR, KBB:K, KNB:K and so on). For example in case of king, bishop and knight against king, there are at least 2 different ways (W-system and some triangle thing, see [MuLa03]). These optimal strategies provide polynomial-time algorithms and should be described in a strictly mathematical way.

Besides an understanding of fortresses (drawn endgame positions where the side lagging in material sets up a zone of protection around the king) would be desirable. Computers are unable to reason about fortress-type positions (without tablebases).

Perhaps to investigate the Lie algebra structure is all you need for an optimal strategy. Or it may be a polynomial task to find out about the optimal moves, whereas it remains unknown und undecidable at this point if the given position is won, drawn or lost. So this wouldn't be a perfect solution, but it would be sufficient for a strong engine and would have advantages over traditional game tree search.

Last but not least we cite a motivational remark taken from [182] (describing a certain model for the moving of Chess pieces):

> [. . .]. The Chess move operators can be encoded by a group-equivariant matrix; rapid multiplication of a group-equivariant matrix by a vector, in general, relies on (decomposability:) the algebra-isomorphism between a group algebra and a (direct in the context of using integers) sum of matrix algebras [. . .].
>
> — Lewis Benjamin Stiller

Well, dreaming is allowed :-)

Chapter 20

Resume

20.1 What's next?

See [139] for the following statement:

> [. . .]. I often use the analogy of a Chess game: one can learn all the rules of Chess, but one doesn't know how to play well. [. . .] The present situation in physics is as if we know Chess, but we don't know one or two rules. [. . .] In Chess, bringing the pieces towards the centre of the board increases their general strength. This is a principle which is not contained explicitly in reading the rules of Chess, but can be understood in terms of the rules of Chess in an indirect fashion. The principle is obviously a consequence of only those rules and nothing else. [. . .] It's just a question of finding convenient methods for analysing complex systems. [. . .]
>
> — Richard Feynman

This should be enough motivation to start work now!

Part III

Appendix

Here one can find additional information about the installation and background references.

Appendix A

Installation and Usage

Here the installation of XiStrat is described (together with some usage hints). For more detailed information about third party software please consult the respective documentation.

First of all (of course) you need the XiStrat software:

```
tar -zxvf xistrat.tar.gz
```

(or use some other unpacking software). Define an environment variable `XISTRAT_HOME` to point to your installation directory (for example by entering

```
export XISTRAT_HOME = ...
```

on UNIX and

```
set XISTRAT_HOME = ...
```

on Windows).

If not stated otherwise, in what follows the mentioned filenames belong to files residing in the directory where `XISTRAT_HOME` is pointing to.

One possible way to avoid messages like Network is unreachable due to a SocketException (on systems binding only to IPv6 addresses by default) is to explicitely enforce the use of IPv4 as shown below.

A.1 Compiling

Install the Java SDK and Java3D.

To generate the `*.class` files, one may use the Makefile by

```
make clean all
```

. Instead you can also use your favourite IDE, or create a `build.xml` for Ant. As a last resort you might choose the manual approach

```
javac -d ../clss org/xistrat/bean/*.java
```

(directory-wise).

At the moment there is no further selective bundling provided for client or server side deployment (which would make some sense though since not all classes are needed on both sides).

A.2 Gaming

A.2.1 Server

Install the Java JRE and the MySQL server (and client as well for administration).

In order to create and fill the database and the tables (once you have a MySQL server running) start a MySQL client by

```
mysql -p
```

, then set up the tables and privileges (modify the file mentioned below to match your IP addresses and computer names). Per default a database named `theo` and the login `chess/linux` [user/password] is used (from client- and serverside). Be aware that the XiStrat server and clients need TCP/IP access to the MySQL server on port 3306 (pay attention to firewall issues, avoid skip-networking; and for example use bind-address option to localhost only if a local setup is all you need).

```
mysql> \. data/sql/mysql_accounts.sql

mysql> \u theo

mysql> \. data/sql/create_tables.sql
```

Use `bin/csv_import.sh` to import the contents of the `data/dump/*.txt` into the MySQL database. You might create symlinks by

```
cd $MySQL_HOME/data/theo
ln -s $XISTRAT_HOME/data/dump/*.txt .
```

for this purpose. On Windows the `csv_import.bat` script is provided.

The JDBC driver (`com.mysql.jdbc.Driver.class`) is needed at runtime in the `CLASSPATH`, as are the XiStrat class files. Finally,

```
java -Djava.net.preferIPv4Stack=true [-server] \
                    org.xistrat.net.F4Server \
                    [-u] [-p port] [-n games] [-v] [-h]
```

should start the server (per default listening on port 9239) for gaming. The command line options are about enabled undo, the port listening on, maximal number of games, verbose output and help offer.

A.2.2 Client

Install the Java JRE and Java3D.

Whereas you won't have to deal with the (i.e. OpenGL) layer directly, nevertheless some overall settings should be tuned in order to achieve formidable antialiasing etc. (for example in case of some nVidia card):

```
export __GL_FSAA_MODE=4
export __GL_DEFAULT_LOG_ANISO=1
export __GL_SYNC_TO_VBLANK=1
```

Besides put the MySQL JDBC driver (`com.mysql.jdbc.Driver.class`) and the XiStrat class files in your `CLASSPATH`.

```
java -Djava.net.preferIPv4Stack=true \
                    org.xistrat.net.d3.F4Client \
                    [-a host] [-p port] [-m] [-v] [-h]
```

will start a GUI for a human player (for example you might use localhost as the host, this often helps when otherwise you would get a Communications link failure due to a jdbc4.CommunicationsException) (the command line option -m enables additional helping 3D mirror views), whereas

```
java -Djava.net.preferIPv4Stack=true \
             -server -XX:+UseConcMarkSweepGC -XX:NewSize=640m \
             -XX:+RelaxAccessControlCheck -XX:+AggressiveOpts \
             -Xms1280m -Xmx1280m -Xss4096k \
             org.xistrat.net.AutoPlayer \
             -f Chess|Go|Reversi [-a host] [-p port] -g game -n alias -s strength
```

starts an autoplayer computer engine for the respective variant of game. In case of Chess it's thinking strength elementary single move levels (plies) into the future, thereby needing enough heap space for the transposition hashtables. The name of the game to which you want to make contact and the alias with which you want to login correspond to the AWT fields in the human GUI. If you want to let two engines play against each other, first of all start an F4Client and make up a match for them. Of course installing the engine on a big machine will give best results.

A.3 Creating 3D data

The database tables point_pre, point, face_pre and face are used (hardcoded into the source code); those tables are created by

```
mysql> \. data/sql/create_tables_n.sql
```

(comment in/out the correspondent parts in this file, dependent on the graph's face-valence).

Now the tools which are originating from the VRMLGraph project are used.

```
java -Djava.net.preferIPv4Stack=true \
                    org.xistrat.util.d3.CreateData \
                    -g graph [-v variant] [-t trials] [-s 0|1]
```

creates 3D data (into the tables point_pre and face_pre). The mandatory command line option -g graph specifies (without extension or path) an accompanying property file in `data/properties` (containing info about face-valence and number of polygons (incl. invalid ones)) and a similarly named `data/infile/graph[_variant].txt`, and optionally the number of calculations can be set (default is 50000). Besides one can resume a former layouting instead of starting afresh, this is useful in case of lengthy calculation where only some more iterations would have been needed for a perfect result, and it provides a way to suggest a topologically more or less correct layout to start from.

It might be necessary to update the face_pre table in order to mark some faces as being unwanted, or if the first face has got a wrong order of vertices. And in the point_pre table you can manipulate the data as well, for example in order to propose topologically correct locations (instead of random data) for the layouting to start with. In case of quasicrystals directly the [point,face]_quasi_* tables are used since (at least for variants greater than zero) information about the type of the tiles is needed later on (for a subsequent inflation/deflation procedure).

Use

```
java -Djava.net.preferIPv4Stack=true \
                    org.xistrat.util.d3.RefreshData \
                    [-g graph] [-v variant][-c colors][-t trials] [-s 0|1|2]
```

to reset the orientations and colors. The above mentioned number of polygons must be the correct value and it is assumed that the first valid face gives the correct orientation to start with. The command line options are about the number of colors to begin the procedure, how many iterations should be tried, and if the default old-fashioned procedure or the more sophisticated tricky approach should by applied. In the latter case one can start a fresh coloring or can continue a previous run.

Doing the graph variants an inverted orientation can be repaired manually by reflecting one rendered coordinate in 3D, the faces mustn't be touched.

Afterwards some dumping (use the `bin/csv_dump.sh`), renaming the resulting `point_pre.txt` and `face.txt` to the appropriate `point_[graph].txt` and `face_[graph].txt`, adding entries to `board.txt` and `game.txt` within `data/dump`, then adding entries for the new tables point_[graph] and face_[graph] to the file `data/sql/create_tables.sql` and a subsequent import (see the file `bin/csv_import.sh`) should give you another nice little world to play on.

A.4 Knot Viewer

You can visualize knots and links using

```
java -Djava.net.preferIPv4Stack=true \
                    org.xistrat.util.d3.KnotViewer -l linkID [-m]
```

, but this utility is still in its infancy at the moment.

A.5 Export ASCII data

A.5.1 XML persistence

Mainly for some demo applet we provided some alternative to the database import/export.

```
java -Djava.net.preferIPv4Stack=true \
                    org.xistrat.util.d3.XMLGraphExport graph [variant]
```

and

```
java -Djava.net.preferIPv4Stack=true \
                    org.xistrat.util.d3.XMLGraphLoader file
```

Have a look at `*.xml` and `polyhedron.dtd` within `data/xml`. Using the VRML or X3D loaders could have been another possibility.

A.5.2 target GAP export and autogenerating infiles

For example you want to dump some group generators or matrices out of XiStrat, so another software, let's say the computer algebra system GAP, can easily import them. Or you wish to autocreate infiles for graphs resulting from Rubik-like morphing moves or (in the case of quasicrystals) constructed by other rules.

```
java -Djava.net.preferIPv4Stack=true \
                    org.xistrat.util.ExportData \
                    -g graph [-v variant] -o outfile [-s s] \
                    [-m|-q|-rf|-rg|-ra|[-kd|-kp] link] \
                    [-c new_variant] [-f type party role]*
```

The command line options are about the graph (and optionally its variant version to start from), the outfile, telling if flip/rot group generators (is default) (and if using the 'double', fixing labels pointing towards holes, treating holes as valid are used), adjacency matrices of the graph and its dual, holonomy groups (standard translation, with proj, local rot, relations, covers or atlas), morphing-autogenerated infiles, cross groups, the full morph group, some knot data or polynomials, or a new quasicrystals iteration are wanted, or optionally some (Chess) pieces can be specified (but at the moment most options are only usable without this). It seems that this vast amount of command line arguments could be arranged somehow in a cleaner way?!

Import the exported data into GAP by

```
gap> Read(infile);
```

And now you can play with it (examples are in Chapter 7). See references for further information.

Appendix B

Third Party Software

Category	
OS	GNU/Linux-2.6.30+
	Windows, Solaris, Mac OS X
database	MySQL-5.1.39
JDBC driver	Connector/J-5.1.10
Java SDK	Sun-1.7.0
	IBM-1.6.0
Java3D	-1.5.2
Java native	GCJ-4.3.3
X Window System	X.org-7.4
IDE	XEmacs-21.4.21 / JDE-2.3.5.1
image manipulation / screenshot grabber	The Gimp-2.6
web browser	Firefox-3.0
DTD	DocBook XML V5.0-extension MathML-2.0
XSL stylesheets	DocBook-XSL-NS-1.76.0 / DBLaTeX-0.3
XSLT processor	XSLTProc-1.1.18 / Xalan-2.7.1
(pdf)eTeX	TeXLive-2009 / pdfeTeX-1.40.10
computer algebra program	GAP-4.4.12
	KANT/KASH-3.0
	Singular-3.1.0

Table B.1: *Software*

Appendix C

Bibliography

C.1 Books, Articles, URLs

Here are several interesting software projects as well as a wide collection of literature for further reading. Well, by now we have cited the whole local library.

C.1.1 Preferences

C.1.1.1 General Mathematics

[1] Front for the Mathematics ArXiv.
URL.

[2] Notices / Bulletin of the AMS.
URL.
URL.

[3] Mathematics Preprint Search System.
URL.

[4] Mathworld Wolfram.
URL.

[5] D. Rusin, Index via Mathematics Subject Classification.
URL.

[6] Google Scholar.
URL.

[7] Wikipedia.
URL.
URL.
URL.
URL.
URL.
URL.

C.1.1.2 Graph Theory and Combinatorics

[8] P. J. Cameron, "Combinatorics: Topics, Techniques, Algorithms", Cambridge University Press, Copyright © 1994 .

[9] The Electric Journal of Combinatorics.
URL.

[10] G. Di Battista, P. Eades, and I. G. Tollis, "Graph Drawing", Prentice-Hall, Copyright © 1999 .

[11] Graduate Texts in Mathematics, 173, 3.
R. Diestel, "Graph Theory", Springer Verlag, Copyright © 2005 .

[12] R. L. Graham, M. Grötschel, and L. Lovász, "Handbook of Combinatorics I + II", Elsevier, Copyright © 1995 .

[13] T. R. Jensen and B. Toft, "Graph Coloring Problems", Wiley, Copyright © 1995 .

[14] J. Comp. Syst. Sci., 25.
E. M. Luks, "Isomorphism of graphs of bounded valence can be tested in polynomial time", Copyright © 1982 , 42-65.

[15] Computers & Graphics, Elsevier.
M. McClure, "A Stochastic Cellular Automaton for Three-Coloring Penrose Tiles", Copyright © 2001 .
URL.

[16] B. Mohar and C. Thomassen, "Graphs on Surfaces", Hohns Hopkins University Press, Copyright © 2001 .
URL.

[17] M. Molloy and B. Reed, "Graph Colouring and the Probabilistic Method", Springer Verlag, Copyright © 2001 .

[18] B. D. McKay, The Nauty Page.
URL.

[19] V. Remie, "Graph isomorphism problem", Backelor Project, Copyright © 1997 , Eindhoven University of Technology.
URL.

[20] Sloane's On-Line Encyclopedia of Integer Sequences, Copyright © .
URL.

[21] Journal of combinatorial theory, Series B 62.
C. Thomassen, "Five-Coloring Graphs on the Torus", Copyright © 1994 , 11-33.

[22] Transactions of the americ. mathematical society, 349.
C. Thomassen and J. Gimbel, "Coloring Graphs with fixed genus and girth", Copyright © 1997 , 4555-4564.

[23] W. T. Tutte, "Graph Theory", Copyright © 2001 .

[24] L. W. Beineke and R. J. Wilson, Selected Topics in Graph Theory, Copyright © 1978 .
Wilson et. al, "Eigenvalues".

[25] L. W. Beineke, R. J. Wilson, and P. J. Cameron, Topics in Algebraic Graph Theory, Cambridge University Press, Copyright © 2004 .
L. H. Soicher, "Computing with graphs and groups".
URL.

C.1.1.3 Topology

[26] Advanced Courses in Mathematics.

W. G. Dwyer and H.-W. Henn, "Homotopy Theoretic Methods in Group Cohomology", Copyright © 2001 , Birkhäuser.

[27] Graduate Texts in Mathematics, 153.

W. Fulton, "Algebraic Topology", A First Course, Copyright © 1995 , Springer Verlag.

[28] P. J. Giblin, "Graphs, Surfaces and Homology", Copyright © 1977 , Chapman and Hall.

[29] W. Greub, S. Halperin, and R. Vanstone, "Connections, Curvature and Cohomology", I, II, III, Copyright © 1976 , Academic Press.

[30] A. Hatcher, "Algebraic Topology", Copyright © 2002 , Cambridge University Press.

URL.

[31] R. V. Gamkrelidze and V. A. Vassiliev, Encyclopaedia of Mathematical Sciences, 141, Low-Dimensional Topology II, Copyright © 2004 Springer Verlag.

S. K. Lando and A. K. Zvonkin, "Graphs on Surfaces and their Applications".

[32] D. Martin, "Manifold Theory", an introduction for mathematical physicists, Copyright © 1991 , Ellis Horwood Limited.

[33] Graduate Texts in Mathematics, 127.

W. S. Massey, "A Basic Course in Algebraic Topology", Copyright © 1991 , Springer Verlag.

[34] Graduate Texts in Mathematics, 72.

J. Stillwell, "Classical Topology and Combinatorial Group Theory", Copyright © 1980 , Springer Verlag.

C.1.1.4 Knots Theory

[35] C. C. Adams, "The Knot Book", W. H. Freeman and Company, New York, Copyright © 1994 .

"Das Knotenbuch", Spektrum Verlag, Copyright © 1995 .

[36] R. G. Scharein, The KnotPlot Site.

URL.

[37] H. R. Morton, Liverpool University Knot Theory Group, Programs and Procedures.

URL.

[38] P. Suber, Knots on the Web.

URL.

[39] M. Thistlethwaite, KnotScape.

URL.

[40] V. O. Manturov, "Knot Theory", Chapman and Hall / CRC, Copyright © 2004 .

[41] B. Sanderson, The Knot Theory MA3F2 Page.

URL.

[42] S. V. Jablan, New Knot Tables.

URL.

C.1.1.5 Group Theory, Algebra and Representations

[43] D. Joyner, R. Kreminski, and J. Turisco, Applied Abstract Algebra, Copyright © 2002 .
URL.

[44] Grundlehre der mathematischen Wissenschaften, 309, 2.
A. Adem and R. J. Milgram, "Cohomology of Finite Groups", Copyright © 2004 , Springer Verlag.

[45] M. Aschbacher, "Finite group theory", Copyright © 1986 , Cambridge University Press.

[46] M. Aschbacher, "Sporadic Groups", Copyright © 1994 , Cambridge University Press.

[47] M. Aschbacher, "The Status of the Classification of the Finite Simple Groups", Copyright © 2004 , Notices of the AMS.
URL.

[48] ATLAS of Finite Group Representations.
URL.

[49] D. J. Benson, "Groups, Representations and Cohomology Preprint Archive", Copyright © 2005 .
URL.

[50] R. Boltje, "Math 214: Theory of Finite Groups", Copyright © 1999 , University of California.
URL.

[51] London Mathematical Society Lecture Notes Series.
Th. Breuer, "Characters and Automorphism Groups of Compact Riemann Surfaces", Copyright © 2000 , Cambridge University Press.
URL.

[52] P. J. Cameron, Permutation Groups, Copyright © 1999 , Cambridge University Press.
URL.

[53] P. J. Cameron, Classical Groups, Copyright © 2000 , Lecture Notes.
URL.

[54] D. F. Holt, COHOMOLO GAP package, Cohomology groups of finite groups on finite modules, Copyright © 2003 .
URL.

[55] G. Hiss, Chr. Jansen, K. Lux, and R. Parker, Computational Modular Character Theory, Copyright © 1992 .
URL.

[56] C. W. Curtis and I. Reiner, "Methods of Representation Theory", I, II, Copyright © 1981, 1987 , Wiley.

[57] Graduate Texts in Mathematics, 163.
J. D. Dixon and B. Mortimer, "Permutation Groups", Springer Verlag, Copyright © 1996 .

[58] Graduate Texts in Mathematics, 129.
W. Fulton and J. Harris, "Representation theory", Copyright © 1996 , Springer Verlag.

[59] London Mathematical Society Series.
W. Fulton, "Young Tableaux", Copyright © 1996 , Cambridge University Press.

[60] GAP -- Groups, Algorithms, and Programming, Copyright © 2002 , The GAP Group.

URL.

[61] D. Gorenstein, "Finite Simple Groups (An Introduction to their classification)", Copyright © 1982 , .

[62] L. H. Soicher, The GRAPE package for GAP, Copyright © 2002 .

URL.

URL.

[63] R. Gramlich, "On graphs, geometries, and groups of Lie type", PhD thesis, Copyright © 2002 .

URL.

[64] R. L. Griess, JR., "Twelve Sporadic Groups", Copyright © 1998 , Springer Verlag.

[65] B.C. Hall, "An Elementary Introduction to Groups and Representations".

URL.

[66] G. Ellis, HAP --- Homological Algebra Programming, Copyright © 2005 .

URL.

[67] Discrete Mathematics and Its Applications.

D. F. Holt, B. Eick, and E. A. O'Brien, "Handbook of Computational Group Theory", Copyright © 2005 , CRC Press.

[68] Lecture Notes in Mathematics.

T. Hsu, "Quilts: Central Extensions, Braid Actions, and Finite Groups", Copyright © 2000 , Springer Verlag.

[69] C. Dong and G. Mason, Moonshine, the Monster, and Related Topics, Contemp. Math., Copyright © 1996 AMS, 193, 113-122.

T. Hsu, "Some quilts for the Mathieu groups".

URL.

[70] Experimental Mathematics, 10, 3, 369-381.

A. Hulpke, "Representing subgroups of finitely presented groups by Quotient Subgroups", Copyright © 2001 .

URL.

[71] A. Hulpke, "Constructing transitive permutation groups", Copyright © 2001 .

URL.

[72] A. Hulpke, "Notes on Computational Group Theory", Copyright © 2006 .

URL.

[73] B. Huppert and N. Blackburn, "Finite Groups", I - III, Copyright © 1967 , Springer.

[74] G. James and M. Liebeck, "Representations and Characters of Groups", Cambridge, Copyright © 1993, 2001 .

[75] J. Comput. Math., 8, 122-144, Copyright © 2005 LMS.

Chr. Jansen, "The minimal degrees of faithful representations of the sporadic simple groups and their covering groups".

URL.

[76] D. L. Johnson, "Symmetries", Copyright © 2001 , Springer Verlag.

[77] D. Joyner, "A primer on computational group homology and cohomology", Copyright © 2009 .

URL.

[78] H. Kurzweil and B. Stellmacher, "The Theory of Finite Groups", Copyright © 2004 , Springer Verlag.

[79] V. Bovdi, A. Konovalov, R. Rossmanith, and C. Schneider, LAGUNA GAP package, Lie AlGebras and UNits of group Algebras, Copyright © 2005 .

URL.

[80] Notices of the AMS.

D. K. Maslen and D. N. Rockmore, "The Cooley-Tukey FFT and Group Theory", Copyright © 2001 , AMS.

URL.

[81] H. Bäärnhielm, Schreier-Sims algorithm for matrix groups.

URL.

URL.

[82] Permutation groups resources, Copyright © 2002 , Queen Mary University of London - School of Mathematical Sciences.

URL.

[83] SAGE: Open Source Mathematics Software, Copyright © 2008 , The SAGE Group.

URL.

[84] Á. Seress, "An introduction to Computational Group Theory", Copyright © 1997 , Notices of the AMS.

URL.

[85] Á. Seress, "Permutation Group Algorithms", Copyright © 2003 , Cambridge University Press.

[86] Graduate Texts in Mathematics, 42.

J. P. Serre, "Linear Representations of Finite Groups", Copyright © 1996 , Springer Verlag.

[87] Unifpack (Web Version 2.0), A Package for Studying Riemann Surfaces with Symmetry, Copyright © 1994 , The Geometry Center University of Minnesota.

URL.

C.1.1.6 Number Theory and Lattices

[88] P. E. Bland, "The Basics of Abstract Algebra", Copyright © 2002 , W. H. Freeman and Company, New York.

[89] J. H. Conway and N. J. A. Sloane, "Sphere Packings, Lattices and Groups", Copyright © 2001 , Springer Verlag.

[90] P. Doyle and C. McMullen, Solving the quintic by iteration, Copyright © 1989 .

URL.

[91] Shell of the KANT V4 Software, Copyright © 2003 , The Kant Project.

URL.

[92] R. Lidl and H. Niederreiter, "Introduction to finite fields and their applications", Copyright © 1986, 1994, 1997, 2000 , Cambridge University Press.

C.1.1.7 Algebraic Geometry and Commutative Algebra

[93] D. Bump, "Algebraic Geometry", Copyright © 2001 , World Scientific.

[94] R. V. Gamkrelidze and V. L. Popov, Encyclopaedia of Mathematical Sciences, 130, Invariant Theory and Algebraic Transformation Groups I, Copyright © 2002 Springer Verlag.

H. Derksen and G. Kemper, "Computational Invariant Theory".

[95] G.-M. Greuel, "A Singular introduction to commutative algebra", Springer Verlag, Copyright © 2002 .

[96] M. Kreuzer and L. Robbiano, "Computational Commutative Algebra I", Springer Verlag, Copyright © 2000 .

[97] LMS Student Texts, 12.

M. Reid, "Undergraduate Algebraic Geometry", Copyright © 1988 , Cambridge University Press.

[98] G.-M. Greuel, G. Pfister, and H. Schönemann, A Computer Algebra System for Polynomial Computations, Copyright © 2003 , Centre for Computer Algebra, University of Kaiserslautern.

URL.

[99] Texts and Monographs in Symbolic Computation.

B. Sturmfels, "Algorithms in Invariant Theory", Copyright © 1993 , Springer Verlag.

C.1.1.8 Lie Theory and Geometric Algebra

[100] J. McKay, A Rapid Introduction to ADE Theory.

URL.

[101] Springer Undergraduate Mathematics Series.

A. Baker, "Matrix Groups: An Introduction to Lie Group Theory", Copyright © 2002 , Springer Verlag.

[102] Graduate Texts in Mathematics, 225.

D. Bump, "Lie Groups", Copyright © 2004 , Springer Verlag.

[103] R. W. Carter, "Finite groups of Lie type", Conjugacy classes and complex characters, Copyright © 1985 , Wiley.

[104] M. Çiperlani, The McKay Correspondence, Copyright © 2000 .

URL.

[105] Springer Undergraduate Mathematics Series.

K. Erdmann and M. J. Wildon, "Introduction to Lie Algebras", Copyright © 2006 , Springer Verlag.

[106] J. Fuchs and Chr. Schweigert, "Symmetries, Lie algebras and representations", Copyright © 2000 , Cambridge University Press.

[107] A. W. Knapp, "Lie Groups beyond an introduction", Copyright © 2002 , Birkhäuser Verlag.

[108] J. van Hoboken, "Platonic solids, binary polyhedral groups, Kleinian singularities and Lie algebras of type A, D, E (Master's Thesis)", Copyright © 2002 , University of Amsterdam, Faculty of Mathematics.

URL.

[109] T. Smith, Mathematics.

URL.

URL.

[110] P. Woit, Lie Groups and Representations: Mathematics, Copyright © 2002 .
URL.

C.1.1.9 Polyhedra, Tesselations, Tilings

[111] V. Bulatov's Polyhedra Collection.
URL.

[112] J. Christy, "Pentacontihexahedron", Copyright © .
URL.

[113] G. Egan, "KleinDual", Copyright © 2005 .
URL.

[114] JavaView.
URL.

[115] Polyhedral Computation.
URL.

[116] Polylib.
URL.

[117] Polymake.
URL.

[118] Chr. Pöppe, "Serie zur räumlichen Geometrie", Unendliche Polyeder: Fast Platonische Körper - das wohltemperierte Dodekaeder, Copyright © 1997 - 2004 , Spektrum der Wissenschaften.
URL.

[119] Polytopia: 120 cell.
URL.

[120] The Pharma Polyhedra Library.
URL.

[121] Virtual-Polyhedra.
URL.

C.1.1.10 Noncommutative and Differential Geometry

[122] Differential Geometry (periodic surfaces).
URL.

[123] H. Goenner, Einführung in die spezielle und allgemeine Relativitätstheorie, Copyright © 1996 , Spektrum Akademischer Verlag.

[124] Differential Geometry -- Minimal Surfaces.
URL.

[125] A. Gray, "Modern Differential Geometry (of curves and surfaces with Mathematica)", Copyright © 1998 , CRC-Press, 2.

[126] Scientific Graphics Project.
URL.

[127] Tilings, Hyperbolic Geometry and Computational Group Theory.
URL.

C.1.1.11 QuasiCrystals, Cellular Automata, Difference Equations

[128] Ammann-Beenker tiling.

URL.

[129] Hex Cellular Automaton.

URL.

[130] M. Baake, U. Grimm, and R. V. Moody, "Die verborgene Ordnung der Quasikristalle", Spektrum der Wissenschaften, Copyright © 2002 .

URL.

[131] Cellular Automata.

URL.

URL.

[132] Undergraduate Texts in Mathematics.

P. Cull, M. Flahive, and R. Robson, "Difference Equations", From Rabbits to Chaos, Copyright © 2005 , Springer Verlag.

[133] Undergraduate Texts in Mathematics.

S. N. Elaydi, "An Introduction to Difference Equations", Copyright © 1999 , Springer Verlag, 2.

[134] CA on QuasiCrystal (Penrose).

URL.

[135] CA on triangle or hexagonal lattice.

URL.

[136] J. Conway, "Game of Life".

URLURL.

[137] R. E. Mickens, "Difference Equations", Copyright © 1990 , Van Nostrand.

C.1.1.12 Symmetry Theory, Mathematical Physics

[138] G. W. Bluman and S. Kumei, Symmetries and Differential Equations, Copyright © 1989 , Springer Verlag.

[139] P. C. W. Davies and J. R. Brown, SuperStrings - A Theory of Everything?, Copyright © 1988 , Cambridge University Press.

[140] P. E. Hydon, Symmetry Methods for Differential Equations: A Beginner's Guide, Copyright © 2000 , Cambridge University Press.

[141] P. J. Olver, Applications of Lie Groups to Differential Equations, Copyright © 1986 , Springer Verlag.

[142] P. J. Olver, Equivalence, Invariants and Symmetry, Copyright © 1995 , Cambridge University Press.

[143] H. Stephani, Differential Equations - Their solution using symmetries, Copyright © 1989 , Cambridge University Press.

[144] P. G. Doyle and J. L. Snell, Random walks and electric networks, Copyright © 2000 .

URL.

C.1.1.13 Complexity Theory

[145] ClayMath Millenium Prizes.
URL.

[146] M. R. Garey and D. S. Johnson, "Computers and Intractability; a Guide to the theory of NP-completeness", Copyright © 1979 .

[147] H. R. Lewis and Ch. H. Papadimitriou, *The efficiency of Algorithms*, 238, Sci. American, 1978, Copyright © 1978 .

[148] R. Uehara, "Probabilistic Algorithms and Complexity Classes", Copyright © 1998 The University of Electro-Communications.
URL.

C.1.1.14 Rubik's Cube

[149] Rubik's Cube solver.
URL.

[150] 4D Magic Rubik's Cube.
URL.

[151] Jaap's Puzzle Page.
URL.

[152] D. Joyner, Mathematics of the Rubik's Cube, Chess and some papers.
URL.
URL.
URL.
URL.

[153] J. Jelinek and H. M. Dr. Bizek, The Rubik's Cube - Methods of Solution.
URL.

[154] M. Monroe, Solving the Rubik's Cube.
URL.

[155] L. Petrus, Rubik's Cube Solution.
URL.

[156] G. Helm, Rubik's Cube The Ultimate Solution.
URL.

[157] M. Reid, Michael Reid's Rubik's Cube page.
URL.

[158] T. Rokicki, "Twenty-Five Moves Suffice for Rubik's Cube".
URL.

[159] D. Byrden, Twisty puzzles.
URL.

C.1.1.15 Games and Puzzles

[160] Abstract Games Magazine.
URL.

[161] University of Alberta Computing Science Technical Report TR94-08 .
J. C. Culberson and J. Schaeffer, "Efficiently Searching the 15-Puzzle", Copyright © 1994 .
URL.
URL.

[162] Proc. of the Int'l Symposium on Symbolic and Algebraic Computing (ISSAC).
S. Egner and M. Püschel, "Solving Puzzles related to Permutation Groups", Copyright © 1998 .
URL.

[163] R. E. Korf and L. A. Taylor, Finding optimal solutions to the Twenty-Four Puzzle, Copyright © 1996 .
URL.

[164] Knight's Tour Notes.
URL.

[165] Loyd's 15-16 Puzzle.
URL.

[166] Conway's M13.
URL.

[167] Ed Pegg's Puzzle Page.
URL.

[168] N Queens without backtracking search.
URL.
URL.
URL.

[169] E. W. Weisstein, non-attacking kings, From MathWorld -- A Wolfram Web Resource.
URL.
URL.

C.1.1.16 Game Theory, Engines, Software

[170] L. V. Allis, H. J. van den Herik, and M. P. H. Huntjens, Go-Moku and Threat-Space Search, Copyright © 1993 .

[171] Proc. of the AAAI.
L. V. Allis, H. J. van den Herik, and M. P. H. Huntjens, "Go-Moku Solved by New Search Techniques", Copyright © 1993 .

[172] Arimaa, The game of real intelligence.
URL.

[173] E. R. Berlekamp, "The Dots and Boxes Game", Sophisticated Child's Play, A K Peters, Copyright © 2000 .
URL.

[174] J. Bewersdorff, "Luck, Logic, and White Lies ", The Mathematics of Games, A K Peters, Copyright © 2004 .
URL.

[175] C. Browne, "Connection Games", Variations on a Theme, A K Peters, Copyright © 2005 .
URL.

[176] Buggy the draughts program.
URL.

[177] Chinook.
URL.

[178] Gambit Game Theory Analysis Software and Tools.
URL.

[179] M. Trautmann and F. Fischer, Zèrtz game software.
URL.

[180] JHex.
URL.

[181] The Moku Family.
URL.

[182] MSRI Publications, 29, Cambridge University Press.
R. J. Nowakowski, Games of No Chance, Copyright © 1996 .
URL.

[183] AAAI: Proceedings.
URL.
D. S. Nau, "Pathology on Game Trees: A Summary of Results", Copyright © 1980 .
URL.

[184] Artificial Intelligence, 21.
D. S. Nau, "Pathology on Game Trees revisited and an alternative to minimaxing", Copyright © 1983 .

[185] Octi online.
URL.

[186] T. B. Laursen, Overflow - The Game.
URL.

[187] Pente.
URL.

[188] Research Series, 117, Tinbergen Institute.
A. Plaat, "Research Re: search & Re-search", PhD thesis, Copyright © 1996 .

[189] Online Renju Class.
URL.

[190] J. van Rijswijck, "Computer Hex: Are Bees better than Fruitflies?", M.Sc. thesis, Copyright © 2000 .
URL.
URL.

[191] Samuel - An open source Draughts Game for Linux.
URL.

[192] A Hex playing program.
URL.

[193] Stahlfaust - Gomoku AI player.
URL.

[194] TwixT fanatic.
URL.

[195] P. Ulfhake, "a Chinese Checkers playing program", Copyright © 1999, 2000 .
URL.

[196] R. Klaehn, HalmaGame, Copyright © 2000 .
URL.

[197] Chinese Checkers - computer games, rules and history.
URL.

C.1.1.17 Reversi, Othello

[198] Othello Engines.
URLURL.

[199] Darwersi (a Darwinian Reversi).
URL.

[200] J2ME Reversi.
URL.

[201] T. Landau, "Othello: Brief & Basic", An introduction to strategy and tactics for the game of Othello (Official handbook of the United States Othello Association), Copyright © 1987 .

[202] M. Buro, "Logistello".
URL.

[203] Othello Opening Strategy.
URL.
URL.

[204] Rhino Othello/Reversi Program.
URL.

[205] Zebra Othello Program.
URL.

C.1.1.18 Go

[206] D. Carlton, "Annotated Go Bibliographies", Copyright © 2002 .
URL.

[207] "Go Bibliographies".
URL.

[208] "3d Battle Go".
URL.

[209] E. R. Berlekamp and D. Wolfe, "Mathematical Go", Chilling Gets the Last Point, A K Peters, Copyright © 1994 .
URL.

[210] "CGOS - A 9x9 Computer Go Server".
URL.

[211] "Combinatorial Game Suite".
URL.

[212] Combinatorial Game Theory.
URL.
URL.
URL.

[213] "Freed Go and Gox", The Game of Go in 3D / Hexagonally.
URL.
URL.

[214] Fuego.
URL.

[215] Go Variants.
URL.

[216] "Commented Bestiary".
URL.

[217] "Go Base".
URL.

[218] Gogo (Go variant on a torus).
URL.

[219] GoGui.
URL.

[220] GNU Go.
URL.

[221] Gozilla.
URL.

[222] "Go News & Topics".
URL.

[223] "Go4Go.net -- Promoting Go (WeiQi, Baduk)".
URL.

[224] "GoGrinder".
URL.

[225] "Go Game World".
URL.

[226] "GnomeGo GUI".
URL.

[227] "Go Problems".
URL.

[228] "Go Teaching Ladder".
URL.

[229] "HandyGo - J2ME Weiqi/Go Client".
URL.

[230] "JimmyGo - J2ME SGF Viewer".
URL.

[231] "Go related bookmarks".
URL.
URL.

[232] R. Jasiek, "Go (Weiqi, Baduk) Rules".
URL.

[233] J. Mattern, "Go - der richtige Zeitpunkt", Copyright © 1991 , Carola-Wild-Verlag, Kassel.

[234] "mgo - The Game of Go on Mobile Phones".
URL.

[235] Computer Go / NeuroGo.
URL.
URL.

[236] "Orego".
URL.

[237] "Sensei's Go Library", (Wiki Wiki WebSite).
URL.

[238] "Tsumego Homepage".
URL.

[239] U. Goertz, "u-go.net".
URL.

[240] M. Reiss, Go Game Software.
URL.
URL.

[241] ICGA, 26.
E. C. D. van der Werf, H. J. van den Herik, and J. W. H. M. Uiterwijk, "Solving Go on Small Boards", Copyright © 2003 .
URL.

[242] G. Seigen, "Go on Go: The analyzed games".
S. Michiharu, "How to play against stronger players (I + II)".
URL.
URL.

[243] "Daily Yomiuri Online".
URL.

C.1.1.19 Chess

[244] G. Bakcsi, "Ungarische Schachproblem Anthologie", Copyright © 1983 .

[245] "Chess Problems".
URL.

[246] Fairy Chess variants.
URL.

[247] Computer Chess.
URL.

[248] R. M. Hyatt, Crafty.
URL.
URL.

[249] R. M. Hyatt and A. E. Cozzie, The effect of hash collisions in a Computer Chess program.
URL.

[250] K. Fabel, "Kurioses Schach", W. Rau Verlag, Copyright © 1972 .

[251] K. Fabel, "Einführung in das Problemschach", Schachverlag R. Schmaus, Copyright © 1976 .

[252] E. Gik, "Faszination Computerschach", W. Rau Verlag, Copyright © 1991 .

[253] E. Gik and A. Karpow, "Schach Studien", Beyer Verlag, Copyright © 1991 .

[254] GNU Chess.
URL.

[255] E. A. Heinz, "Scalable Search in Computer Chess", Algorithmic Enhancements and Experiments at High Search Depths, Vieweg Verlag, Copyright © 2000 .
URL.

[256] A. Karpow and J. Gik, "Schach-Studien", Beyer Verlag, Copyright © 1991 .

[257] H. Kasparjan, "Zauber des Endspiels", W. Rau Verlag, Copyright © 1974, 1985 .

[258] A. Kraemer and E. Zepler, "Im Banne des Schachproblems", de Gruyter, Copyright © 1982 .

[259] Phalanx Chess engine.
URL.

[260] W. Speckmann, "Das logische Schachproblem", W. Rau Verlag, Copyright © 1980 .

[261] W. Speckmann, "Strategie im Schachproblem", de Gruyter, Copyright © 1982 .

[262] W. Speckmann, "Perlen der Schachkomposition (3-Züger)", de Gruyter, Copyright © 1985 .

[263] Schwalbe HomePage.
URL.

[264] Shredder Endgame Database.
URL.

[265] Sjeng: a chess-and-variants playing program.
URL.

[266] Xiangqi-Engine (Chinese Chess).
URL.

[267] WBEC Ridderkerk.
URL.

[268] Fairy Chess Piece.
URL.

[MuLa03] K. Müller and F. Lamprecht, "Grundlagen der Schachendspiele", Gambit, Copyright © 2003 .

C.1.1.20 Online Games Sites, Clients, Editors, Clubs, Events

[269] BoardSpace, A Place for Board Games.
URL.

[270] Computer-Go.
URL.

[271] ChessGames, Online Chess Games Database.
URL.

[272] Go (IGS) client.
URL.

[273] Computer Go Forum, WCGC Gifu Challenge.
URL.

[274] CGoban.
URL.
URL.

[275] Dweebo's Stone Games.
URL.

[276] Eboard Chess board interface.
URL.

[277] Free Internet Chess Server.
URL.

[278] gGo Java Client and SGF editor.
URL.

[279] Go Text Protocol.
URL.

[280] "Go Ratings of Professional and Strong Amateur Players".
URL.

[281] ICGA.
URL.
URL.

[282] Internet Go Server.
URL.

[283] "Intelligent Go Foundation".
URL.

[284] ItsYourTurn.
URL.

[285] JaGo Java Client and SGF editor.
URL.

[286] Java client for Chess servers.
URL.

[287] Java based Interface for Go.
URL.

[288] KGS Go Server.
URL.

[289] Little Golem, online board games.
URL.

[290] MindSports ArenA.
URL.

[291] "MyChess.com - Chess to enjoy".
URL.

[292] Nihon Ki-in (Japanese Go Association).
URL.

[293] NN Go Server.
URL.

[294] Computer Olympiad (2001-2008), WCCC.
URL.

[295] Quarry.
URL.

[296] RedHotPawn.com - Play free online chess.
URL.

[297] Scid - A Chess Database Application.
URL.

[298] The Week in Chess.
URL.

[299] XBoard.
URL.

[300] Yahoo Board Games.
URL.

[301] Weiqi Webring.
URL.

[MSO] "Mind Sports Olympiad Worldwide".
URL.

C.1.1.21 XML, JAVA, 3D

[302] DocBook.
URL.
URL.

[303] DBLaTeX.
URL.

[304] W. Gamba, Forza4, (source free to use and redistribute, please state origin, commerce must be asked about).
URL.

[305] FlyingGuns - A Distributed 3D Simulation Framework.
URL.

[306] Sun Java and Java3D.
URL.
URL.

[307] Java3D.
URL.

[308] J3D.ORG - Tutorials, Resources, Source Code, Links, Books and more for the beginner and advanced Java 3D programmer.
URL.

[309] Java bindings for OpenGL.
URL.

[310] Mobile 3D Graphics API for J2ME.
URL.
URL.
URL.

[311] Mathematical Markup Language.
URL.
URL.

[312] Nonblocking I/O for Java.
URL.
URL.

[313] Noise and Turbulence (Procedural Textures).
URL.

[314] P. J. Mutton, VRMLGraph, LGPL.
URL.

[315] Xalan-Java XSLT processor.
URL.

[316] High performance scenegraph and renderer for gaming.
URL.
URL.
URL.

C.2 Usenet

The following newsgroups may offer more information on various topics.

C.2.1 Preferences

[comp.graphics] Computer Graphics.
news:comp.graphics.algorithms.
news:comp.graphics.visualization.

[comp.theory] Theoretical Computer Science.
news:comp.theory.
news:comp.theory.cell-automata.

[rec.games] Games.
news:rec.games.abstract.
news:rec.games.chess.computer.
news:rec.games.go.

[sci.math] Mathematics.
news:sci.math.research.
news:sci.math.symbolic.

Appendix D

GNU Free Documentation License

Version 1.3, 3 November 2008

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright (C) YEAR YOUR NAME.
>
> Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

> with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Appendix E

Glossary

E.1 What's up?

Here frequently used terms are explained.

C

Complexity (`Complexity`)

Some things can be verified in polynomial time, but cannot be found without exponential effort.

E

Extensible Markup Language (`XML`)

Way to represent contents. See Also "Standard Generalized Markup Language".

S

SGML

See "Standard Generalized Markup Language".

Standard Generalized Markup Language (`SGML`) [ISO 8879:1986]

Some reasonable definition here. See Also "Extensible Markup Language".

X

eXtended STRATegy (`XiStrat`)

The name of this cool software.

Index