

Duale Kegel und Automorphismengruppen hyperbolischer Gitter

vorgelegt als Masterarbeit
der Fakultät für Mathematik, Informatik und
Naturwissenschaften
der Rheinisch-Westfälischen Technischen Hochschule
Aachen

Gutachter: Prof. Dr. G. Nebe, Lehrstuhl D für Mathematik
Prof. Dr. W. Plesken, Lehrstuhl B für Mathematik

Michael H. Mertens (B.Sc.)

September 2012

Inhaltsverzeichnis

1	Einleitung	5
2	Gitter	9
2.1	Allgemeines	9
2.2	Der WATSON-Prozess	12
2.3	Algorithmen für positiv definite Gitter	14
2.4	Der VORONOI-Algorithmus	16
3	Duale Kegel	21
3.1	Kegel und perfekte Punkte	21
3.2	Diskontinuierliche Gruppen	28
3.3	Fundamentalbereiche und Endlichkeitsaussagen	33
4	Der Normalisator-Algorithmus	37
4.1	BRAVAIS-Gruppen	37
4.2	Der Algorithmus	39
5	Automorphismengruppen	41
5.1	Hyperbolische Gitter und duale Kegel	41
5.2	Zulässigkeit	43
5.3	Kürzeste Vektoren und Automorphismen	45
5.4	Reflektive Gruppen	49
5.5	Endlichkeit des Restklassengraphen	50
5.6	Hyperbolische Gitter und der WATSON-Prozess	52
6	Beispiele	53
6.1	Die Automorphismengruppen zu einigen hyperbolischen Gittern	53
6.1.1	Standardform	53
6.1.2	Andere Formen	59
A	Grundlegende Definitionen aus der Graphentheorie	69
B	Magma-Code	73
B.1	Der Automorphismen-Algorithmus	73
B.2	Erläuterung der Intrinsic	95
B.2.1	Das Automorphismen-Paket	95
	Symbolverzeichnis	97
	Symbolverzeichnis	99

Stichwortverzeichnis	100
Literaturverzeichnis	105
Eigenständigkeitserklärung	107
Eigenständigkeitserklärung	107

Kapitel 1

Einleitung

Im Jahre 1908 veröffentlichte G.F. VORONOI einen seiner bedeutendsten Artikel *Nouvelles applications des paramètres continus à la théorie des formes quadratiques* [Vor08], in welchem er den nach ihm benannten VORONOI-Algorithmus zur Berechnung aller perfekten quadratischen Formen mit gegebener Dimension und gegebenem Minimum entwickelt. M. KOECHER verallgemeinerte VORONOI's Methoden mithilfe von selbstdualen Kegeln (vgl. [Koe57] und [Koe60], dort bezeichnet als Positivitätsbereiche) um so Erzeuger bestimmter arithmetischer Gruppen zu finden. Später konnte J. OPGENORTH unter Verwendung einer weiteren Verallgemeinerung der KOECHERSchen Theorie auf duale Kegel (siehe [Opg96], [OPS98] und [Opg01]) hieraus ein Konzept entwickeln, aus dem sich u.a. Algorithmen zur Berechnung von Normalisatoren endlicher unimodularer Gruppen sowie Automorphismengruppen hyperbolischer Gitter ableiten lassen, wobei letzterer bisher nur in einer unveröffentlichten Version von [Opg01] zu finden ist.

Normalisatoren endlicher unimodularer Gruppen treten z.B. in der Kristallographie auf. Eine wichtige Fragestellung dort ist die Klassifikation so genannter kristallographischer Raumgruppen, also gewisser endlicher Erweiterungen freier abelscher Gruppen. Eines der zentralen Resultate in diesem Zusammenhang ist der Satz von BIEBERBACH, FROBENIUS und ZASSENHAUS, demgemäß die Menge der Isomorphieklassen aller Erweiterungen von \mathbb{Z}^n mit einer endlichen Gruppe $G \leq \mathrm{GL}_n(\mathbb{Z})$ in Bijektion zur Menge der $N_{\mathbb{Z}}(G)$ -Bahnen der ersten Kohomologiegruppe $H^1(G, \mathbb{Q}^n/\mathbb{Z}^n)$ steht (siehe [Ple] für Details). Um dies algorithmisch zu handhaben ist es also notwendig, Erzeuger für $N_{\mathbb{Z}}(G)$ bestimmen zu können. Dies ist zum Beispiel im Computer-Paket CARAT ([OPS98]) realisiert.

Hyperbolische Gitter finden ebenfalls in diversen Bereichen Anwendung. So ist beispielsweise eine Konstruktion des berühmten LEECH-Gitters Λ , die von J.H. CONWAY erdacht wurde. Demnach ist Λ als orthogonales Komplement des Norm-0-Vektors $x = (1, 2, \dots, 24, 70)$ im hyperbolischen Gitter zur Form $O_{24,1} = \mathrm{diag}(1, \dots, 1, -1)$ enthalten. Durch diese Beobachtung war CONWAY zudem in der Lage, die Automorphismengruppe von Λ als Untergruppe in der Automorphismengruppe von $O_{24,1}$ wiederzufinden. Den maximalen Normalteiler C_{O_1} vom Index 2 in $\mathrm{Aut}(\Lambda)$ nennt man daher auch die erste CONWAY-Gruppe. Weitere Resultate zu hyperbolischen Gittern und ihren Automorphismengruppen finden sich u.a. auch bei E. VINBERG und R. BORCHERDS, wobei

dort immer der Standardvertreter hyperbolischer Gitter betrachtet wird. Einen allgemeinen algorithmischen Zugang gab es hierzu allerdings nur bedingt.

Die grundlegende Idee hinter den hier präsentierten Algorithmen ist die folgende: Eine Gruppe Ω operiere eigentlich diskontinuierlich (vgl. Definition 3.2.1) auf einem der beiden dualen Kegel. Man überdeckt einen solchen Kegel mit bestimmten Pyramiden, die von so genannten D -perfekten Punkten induziert werden, den so genannten D -VORONOI-Bereichen (vgl. Definition 3.1.4) (KOECHER spricht hier von vollkommenen Pyramiden). Unter gewissen Bedingungen ist diese Kachelung exakt (vgl. Satz 3.2.10), d.h. die Pyramiden schneiden sich höchstens an ihren Seitenflächen. Aus den Nachbarschaftsbeziehungen dieser Pyramiden in einem Vertretersystem unter der Operation von Ω lässt sich dann ein Erzeugendensystem der Gruppe Ω mittels eines Resultates von BASS und SERRE rekonstruieren. Das so konstruierte Erzeugendensystem ist genau dann endlich, wenn es ein endliches Vertretersystem der oben erwähnten Pyramiden gibt.

Hauptziel dieser Arbeit soll sein die von OPGENORTH gefundenen Resultate und Algorithmen zu präsentieren und zu ergänzen. OPGENORTH konnte zeigen, dass der von ihm gefundene Normalisatoralgorithmus immer terminiert. Im Falle der hyperbolischen Gitter ist dies noch immer eine offene Frage, wobei in Proposition 5.5.1 zumindest eine hinreichende Bedingung dafür angegeben wird, dass der Algorithmus terminiert.

Die Arbeit ist folgendermaßen organisiert: In Kapitel 2 wird eine kurze Übersicht allgemeiner Resultate über Gitter gegeben mit dem Ziel, den (klassischen) VORONOI-Algorithmus zu beschreiben. Kapitel 3 behandelt das theoretische Gerüst der dualen Kegel, aus dem in den Kapiteln 4 und 5 die bereits erwähnten Algorithmen als Anwendungen vorgestellt werden. Schließlich werden in Kapitel 6 diverse Beispiele zum Automorphismen-Algorithmus präsentiert. Für die hierzu notwendigen Berechnungen wurde die schon bestehende Implementierung des Normalisatoralgorithmus im Computeralgebrasystem **Magma** (siehe [BCP97]) von M. KIRSCHMER (basierend auf [OPS98]) von mir überarbeitet und ergänzt, sowie der Automorphismenalgorithmus implementiert. Im Anhang A findet sich zudem ein kurzes Kapitel, in dem die hier wichtigsten Begriffe und Definitionen aus der Graphentheorie präsentiert werden. Anhang B enthält schließlich den vollständigen Quellcode des **Magma**-Pakets **AutHyp.m**, sowie eine kurze Anleitung der verfügbaren Funktionen. Das Paket wird demnächst auch online unter

<http://www.math.rwth-aachen.de/~nebe/Bach.html>

verfügbar sein.

Mein besonderer Dank gilt vor allem meiner Betreuerin, Frau Prof. Dr. GABRIELE NEBE, die diese Arbeit angeregt und durch wertvolle Hinweise unterstützt hat, und Herrn Dr. MARKUS KIRSCHMER für seine vielfältige Hilfe und Beratung, auch in Bezug auf die Arbeit mit **Magma**. So hat er mir dankenswerterweise den Quellcode seiner oben erwähnten Implementierung als

Vorlage zur Verfügung gestellt. Desweiteren danke ich Herrn DAVID LORCH dafür, dass er mir für die Arbeit mit dem WATSON-Prozess seine Magma-Implementierung dazu überlassen hat. Herrn Dr. VIKTOR LEVANDOVSKYY möchte ich für seinen Hinweis auf die Theorie der reflektiven Gruppen sowie Herrn Dr. MATTHIAS KÜNZER für seine Hilfe bei der Literaturbeschaffung danken. Nicht zuletzt danke ich meiner Familie, insbesondere meinen Eltern und Großeltern, die mich während des ganzen Studiums in jeder Hinsicht voll und ganz unterstützt haben.

Konventionen Der freie Modul $R^n := R^{1 \times n}$ sei für die gesamte Arbeit als Modul der Zeilenvektoren über einem Ring R verstanden. Jede Koordinatendarstellung eines Vektors ist als Zeile zu interpretieren.

Gruppenoperationen sind hier stets Rechtsoperationen, wenn nicht explizit anders deklariert.

Kapitel 2

Gitter

In diesem Kapitel werden kurz die wesentlichen Begriffe und Resultate über Gitter vorgestellt. Hierbei folgt Abschnitt 2.2 im Wesentlichen [CS88, Chapter 15] für die Erläuterung des Geschlechtssymbols bzw. [Wat62] und [Lor12] für den WATSON-Prozess. Während in Abschnitt 2.1 und Abschnitt 2.2 beliebige Gitter betrachtet werden, sind in Abschnitt 2.3 und Abschnitt 2.4 nur positiv definite Gitter relevant.

2.1 Allgemeines

Zu Beginn geben wir den wohlbekannten SYLVESTERSchen Trägheitssatz für reelle, symmetrische Bilinearformen an, um Missverständnissen mit der Notation vorzubeugen.

Satz 2.1.1. (SYLVESTERScher Trägheitssatz, 1853)

Sei \mathcal{V} ein \mathbb{R} -Vektorraum der Dimension n mit Basis B und $\Phi : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$ eine symmetrische Bilinearform auf \mathcal{V} . Weiter bezeichne $A := {}_B\Phi^B$ die GRAM-Matrix von Φ bezüglich der Basis B . Dann existiert eine Matrix $T \in \text{GL}_n(\mathbb{R})$, so dass

$$TAT^{tr} = \begin{pmatrix} I_p & & \\ & -I_q & \\ & & 0 \end{pmatrix}.$$

Das Tupel $(p, -q)$ ist eindeutig bestimmt und man nennt es den SYLVESTER-Typ oder auch die Signatur von Φ (bzw. von A).

Definition 2.1.2. Sei \mathcal{V} ein reeller Vektorraum der Dimension n mit Basis $B = (b_1, \dots, b_n)$. Eine indefinite, symmetrische Bilinearform $\Phi : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$ mit Signatur $(n-1, -1)$ heißt hyperbolisch.

Mit Φ nennt man auch die GRAM-Matrix

$${}_B\Phi^B := (\Phi(b_i, b_j))_{i,j=1}^n$$

hyperbolisch. Die Menge der reellen, hyperbolischen Matrizen wird mit

$$\mathbb{R}_{hyp}^{n \times n} := \{A \in \mathbb{R}_{sym}^{n \times n} \mid \text{Signatur}(A) = (n-1, -1)\}$$

bezeichnet.

Wir kommen nun zur Definition von Gittern:

Definition 2.1.3. Sei L ein endlich erzeugter, freier \mathbb{Z} -Modul mit Basis $B = (b_1, \dots, b_n)$, ausgestattet mit einer nichtausgearteten, symmetrischen Bilinearform

$$\Phi : L \times L \rightarrow \mathbb{Q}.$$

Dann heißt L ein Gitter und B heißt eine Gitterbasis von L . Gelegentlich wird $\Phi(x, y)$, $x, y \in L$ durch (x, y) abgekürzt. Der Ausdruck $N(x) := \Phi(x, x)$ bezeichne die Norm von $x \in L$.

Man nennt L positiv/negativ definit, hyperbolisch, indefinit, falls die Fortsetzung von Φ auf $\mathcal{V} := \mathbb{R} \otimes_{\mathbb{Z}} L$ diese Eigenschaft besitzt.

Definition 2.1.4. Sei L ein Gitter mit Bilinearform Φ .

(i) Man nennt L ein ganzes Gitter, falls Φ nur ganzzahlige Werte annimmt,

$$\Phi(x, y) \in \mathbb{Z} \text{ für alle } x, y \in L.$$

(ii) Das Gitter

$$L^\# := \{v \in \mathbb{Q} \otimes_{\mathbb{Z}} L \mid \Phi(v, \ell) \in \mathbb{Z} \text{ für alle } \ell \in L\}$$

heißt das zu L duale Gitter.

Offensichtlich ist ein Gitter genau dann ganz, wenn es in seinem dualen Gitter enthalten ist. Somit ist folgende Definition sinnvoll:

Definition 2.1.5. Sei L ein ganzes Gitter bezüglich der Bilinearform Φ . Unter der Diskriminantengruppe $\Delta(L)$ versteht man die Faktorgruppe

$$\Delta(L) := L^\# / L.$$

Ihre Ordnung bezeichnet man als die Diskriminante von L .

Bemerkung 2.1.6. Die Diskriminante eines ganzen Gitters L ist stets endlich. Genauer ist für eine beliebige Basis B von L

$$|\det_B \Phi^B| = |\Delta(L)|.$$

BEWEIS. Folgt im Wesentlichen aus dem Hauptsatz über endlich erzeugte abelsche Gruppen, siehe [Mar03, Proposition 1.1.5] für Details. Die Unabhängigkeit von der Wahl der Basis folgt, da für zwei Basen B, B' von L die für die Basiswechselmatrix gilt $T := {}_{B'}id_B \in \mathrm{GL}_n(\mathbb{Z})$, also $\det T = \pm 1$. Damit gilt

$$\det_{B'} \Phi^{B'} = \det(T_B \Phi^B \cdot T^{tr}) = \det T^2 \cdot \det_B \Phi^B = \det_B \Phi^B. \quad \square$$

Um Gleichheit zweier Gitter im selben Vektorraum zu entscheiden, benötigt man eine kanonische Basis. Diese kann man über die HERMITE-Normalform für ganzzahlige Matrizen erhalten.

Satz 2.1.7. (HERMITE, 1851)

Sei $A \in \mathbb{Z}^{n \times n}$ mit $\det(A) \neq 0$. Dann existiert eine unimodulare Matrix $K \in \text{GL}_n(\mathbb{Z})$ mit $KA = H$, wobei für die Einträge von H gilt:

$$\begin{aligned} h_{ij} &= 0 && , \text{ für } j < i \\ h_{ii} &> 0 && , \text{ für alle } i \\ h_{ij} &\leq 0 \quad \text{und } |h_{ij}| < h_{ii} && , \text{ für } i < j. \end{aligned}$$

Durch diese Bedingungen sind H und K eindeutig bestimmt und man nennt H die HERMITE-Normalform von A .

BEWEIS. [DKT87, Theorem 1.2] □

Definition 2.1.8. (i) Eine Abbildung $\varphi : L_1 \rightarrow L_2$ zwischen zwei Gittern L_1, L_2 mit Bilinearformen Φ_1, Φ_2 heißt eine Isometrie von Gittern, falls φ \mathbb{Z} -linear ist und

$$\Phi_2(\varphi(x), \varphi(y)) = \Phi_1(x, y) \text{ für alle } x, y \in L_1$$

gilt.

(ii) Für ein Gitter L heißt eine bijektive Isometrie $\varphi : L \rightarrow L$ ein Automorphismus von L . Die Gruppe der Automorphismen von L wird mit $\text{Aut}(L)$ bezeichnet.

Bemerkung 2.1.9. (i) Sei B eine Basis des Gitters L mit Bilinearform Φ . Eine \mathbb{Z} -lineare Abbildung $\varphi : L \rightarrow L$ ist dann und nur dann ein Automorphismus von L , wenn ${}_B\varphi_B \in \text{GL}_n(\mathbb{Z})$ und ${}_B\varphi_B \Phi^B \varphi^B = {}_B\Phi^B$ gilt.

(ii) Für positiv (bzw. negativ) definite Gitter ist jede Isometrie injektiv.

BEWEIS. (i): Übersetzt man die Definition eines Automorphismus in Koordinatendarstellung, so folgt die Behauptung.

(ii): Sei $\varphi : L \rightarrow L$ eine Isometrie und $x \in \text{Kern}\varphi$. Dann ist

$$\Phi(x, x) = \Phi(\varphi(x), \varphi(x)) = \Phi(0, 0) = 0.$$

Da Φ definit ist, folgt $x = 0$, also ist φ injektiv. □

Eine stärkere Aussage über Automorphismengruppen definiter Gitter enthält der folgende

Satz 2.1.10. Die Automorphismengruppe eines definiten Gitters ist endlich.

BEWEIS. Ohne Einschränkung sei L ein positiv definites Gitter mit Bilinearform Φ und Basis $B = (b_1, \dots, b_n)$. Vermittels der Basisdarstellung ist $\text{Aut}(L)$ isomorph zu $O({}_B\Phi^B) \cap \text{GL}_n(\mathbb{Z})$ mit $O({}_B\Phi^B) = \{T \in \text{GL}_n(\mathbb{R}) \mid T_B \Phi^B T^{tr} = {}_B\Phi^B\}$. Da Φ positiv definit ist, ist $O({}_B\Phi^B)$ nach Satz 2.1.1 isomorph zu O_n , also insbesondere eine kompakte Gruppe. Die Gruppe $\text{GL}_n(\mathbb{Z})$ ist diskret und der Schnitt einer kompakten mit einer diskreten Menge ist endlich, was zu zeigen war. □

Bemerkung 2.1.11. Es sei darauf hingewiesen, dass Satz 2.1.10 nicht unbedingt richtig bleibt, wenn der umgebende Vektorraum nicht mehr EUKLIDISCH, sondern beispielsweise hyperbolisch ist, denn in diesem Fall ist die orthogonale Gruppe nicht mehr kompakt.

2.2 Der WATSON-Prozess

Definition 2.2.1. Sei L ein Gitter mit Bilinearform Φ . Dann nennt man für jedes $p \in \mathbb{P}$ mit der Eigenschaft, dass $\Delta(L)$ ein Element der Ordnung p^2 enthält, das Gitter

$$\text{Fill}_p(L) := (pL^\# \cap L, \frac{1}{p}\Phi)$$

eine p -Füllung von L .

Bemerkung 2.2.2. Die p -Füllung ist im Allgemeinen nicht durch eine Operation von $\text{GL}_n(\mathbb{Z})$ gegeben, insbesondere sind im Allgemeinen die Automorphismengruppen von L und $\text{Fill}_p(L)$ nicht isomorph.

Allerdings gilt stets

$$\text{Aut}(L) \leq \text{Aut}(\text{Fill}_p(L)) \text{ und } [\text{Aut}(\text{Fill}_p(L)) : \text{Aut}(L)] < \infty.$$

Man kann also $\text{Aut}(L) = \text{Stab}_{\text{Aut}(\text{Fill}_p(L))}(L)$ mittels Bahnenalgorithmus aus $\text{Aut}(\text{Fill}_p(L))$ bestimmen.

In [CS88, Chapter 15] klassifizieren J.H. CONWAY und N.J.A. SLOANE ganzzahlige quadratische Formen und führen dazu eine Notation für das Geschlecht einer quadratischen Form ein:

Definition 2.2.3. Zwei Gitter L, L' liegen im selben Geschlecht, falls

$$\mathbb{R} \otimes L \cong \mathbb{R} \otimes L' \text{ und } \mathbb{Z}_p \otimes L \cong \mathbb{Z}_p \otimes L' \text{ für alle } p \in \mathbb{P}.$$

Hierbei bezeichne \mathbb{Z}_p den Ring der ganzen p -adischen Zahlen, also die Kompletterung von \mathbb{Z} an p .

Motivation für diese Definition ist das so genannte Lokal-Global-Prinzip von HASSE und MINKOWSKI (vgl. [Kne02, Satz 19.1] bzw. [CS88, Chapter 15, Theorem 4]), nach dem zwei quadratische Formen dann und nur dann rational äquivalent sind, wenn sie über \mathbb{R} und jedem \mathbb{Q}_p äquivalent sind. Zwei quadratische Formen mit Matrizen A und $B \in R_{sym}^{n \times n}$ heißen hierbei äquivalent über einem Ring R , falls es eine Matrix $T \in \text{GL}_n(R)$ gibt mit $B = T^{tr}AT$.

Zum Verständnis des Geschlechtssymbols ist weiter noch folgender Satz wichtig:

Satz 2.2.4. (i) Über einem Körper K der Charakteristik $\neq 2$ ist eine quadratische Form mit Matrix A diagonalisierbar, d.h. es gibt eine Matrix $T \in \text{GL}_n(K)$ mit $T^{tr}AT = \text{diag}(a_1, \dots, a_n)$.

(ii) Eine quadratische Form über \mathbb{Z}_p ist für $p \neq 2$ ebenfalls diagonalisierbar.

(iii) Eine quadratische Form über \mathbb{Z}_2 ist äquivalent (über \mathbb{Z}_2) zu einer direkten Summe von quadratischen Formen der Gestalt

$$(qx), \quad \begin{pmatrix} qa & qb \\ qb & qc \end{pmatrix},$$

wo q eine Potenz von 2 ist und es gilt $2 \mid a, c$ und $2 \nmid x, b, ac - b^2$.

BEWEIS. (i) [Kne02, Satz 1.20].

(ii) und (iii): [CS88, Chapter 15, Paragraph 4.4] bzw. [CS88, Chapter 15, Theorem 2]. \square

Korollar 2.2.5. *Jede quadratische Form f über \mathbb{Z}_p lässt sich in eine direkte Summe*

$$f_1 \oplus pf_p \oplus p^2 f_{p^2} \oplus \cdots \oplus qf_q \cdots,$$

zerlegen, wobei die Formen f_q eine zu p teilerfremde Determinante haben (im Falle einer ganzzahligen p -adischen Form) bzw. positiv definit sind (im Falle einer reellen Form). Die Summanden qf_q heißen JORDAN-Konstituenten von f , obige Zerlegung nennt man die JORDAN-Zerlegung von f .

Satz 2.2.6. *Für $p \neq 2$ sei f eine p -adische quadratische Form mit JORDAN-Zerlegung wie in Korollar 2.2.5. Die vorkommenden Werte q (die Stufe von f_q) sowie die Dimension $n_q = \dim f_q$ und das Vorzeichen $\varepsilon_q = \left(\frac{\det f_q}{p}\right)$, wobei letzteres das LEGENDRE-Symbol bezeichnet, bilden ein vollständiges System von Invarianten für f .*

BEWEIS. [CS88, Chapter 15, Theorem 9] \square

Das p -adische Geschlechtssymbol besteht nun aus dem formalen Produkt von Faktoren der Form

$$q^{\varepsilon_q n_q}.$$

Für $p = 2$ gibt es noch einige zusätzliche Invarianten (vgl. [CS88, Chapter 15, Paragraphs 7.3 ff]), die hier jedoch nicht von Bedeutung sind. Im Falle einer reellen Form liefert Satz 2.1.1 ein analoges Resultat.

Bemerkung 2.2.7. *Die Bilinearform Φ eines ganzen Gitters L definiert bekanntermaßen eine quadratische Form f über \mathbb{Z} . Das Geschlecht von L ist nach dem Lokal-Global-Prinzip eindeutig durch die p -adischen Geschlechtssymbole von f (aufgefasst über \mathbb{Z}_p) für $p \mid |\Delta(L)|$ und die Signatur von Φ bestimmt.*

Bemerkung 2.2.8. *Die p -Füllung eines ganzen Gitters L ändert sein Geschlechtssymbol wie folgt:*

Sei die die höchste Stufe eines p -adischen Konstituenten von L mindestens p^2 , was gleichbedeutend damit ist, dass in $\Delta(L)$ ein Element der Ordnung p^2 existiert. Dann reduziert die p -Füllung die Stufe dieses Konstituenten um 2 (vgl. [All12]), ansonsten bleibt das Gitter fest. Führt man nun für jedes $p \mid |\Delta(L)|$ solange eine p -Füllung aus wie möglich, erhält man ein ganzes Gitter, dessen Diskriminantengruppe quadratfreien Exponenten hat. Insbesondere hat das resultierende Gitter eine kleinere Diskriminante als L . Dieses sogenannte WATSON-Gitter von L bezeichnet man mit $\text{Watson}(L)$. Die Abbildung $L \mapsto \text{Watson}(L)$ nennt man auch den WATSON-Prozess.

2.3 Algorithmen für positiv definite Gitter

Für positiv definite Gitter lassen sich einige weitere Begriffe erklären, die im Folgenden noch wichtig sein werden. Von jetzt an sei für den Rest des Kapitels jedes Gitter als positiv definit angenommen, sofern nichts anderes gesagt wird.

In diesem Fall definiert für ein Gitter L das Skalarprodukt Φ eine Norm auf dem EUKLIDISCHEN Vektorraum $\mathcal{V} := \mathbb{R} \otimes_{\mathbb{Z}} L$. Da L eine in \mathcal{V} diskrete Menge ist (vgl. [Mar03, Proposition 1.1.4]) ist folgende Definition sinnvoll:

Definition 2.3.1. *Sei L ein positiv definites Gitter mit Skalarprodukt (\cdot, \cdot) . Dann nennt man*

$$N(L) := \min L := \min_{\{x \in L \setminus \{0\}\}} N(x)$$

das Minimum oder auch die Norm von L . Die Menge der kürzesten Vektoren von L wird mit

$$M(L) := \{x \in L \mid N(x) = \min L\}$$

bezeichnet. Die Kardinalität von $M(L)$ bezeichnet man als die Kusszahl $\kappa(L)$.

Bemerkung 2.3.2. *Wählt man eine Basis B von L , so kann man alle obigen Begriffe auch für die positiv definite Matrix ${}_B\Phi^B$ erklären (mit analogen Bezeichnungen). Für eine beliebige positiv definite Matrix A gilt also*

$$\begin{aligned} \min A &= \min_{x \in \mathbb{Z}^n \setminus \{0\}} xAx^{tr} \\ M(A) &= \{x \in \mathbb{Z}^n \mid xAx^{tr} = \min A\} \\ \kappa(A) &= \#M(A) \end{aligned}$$

Lemma 2.3.3. (HERMITE-Ungleichung)
Jedes Gitter L besitzt eine Basis (b_1, \dots, b_n) mit

$$N(b_1) \cdots N(b_n) \leq \left(\frac{4}{3}\right)^{n(n-1)/2} |\Delta(L)|^2.$$

BEWEIS. [Mar03, Theorem 2.2.1] □

Für die Algorithmen in Kapitel 4 und 5 wird ein essentieller Baustein die Berechnung der kürzesten Vektoren eines Gitters sein. Ein Verfahren hierzu findet sich in [PZ89, Algorithmus (3.15)], der Vollständigkeit halber wird er auch hier kurz beschrieben. Der Algorithmus arbeitet mit der GRAM-Matrix $A = (a_{ij}) \in \mathbb{R}_{sym}^{n \times n}$ des Gitters und berechnet alle Vektoren $x \in \mathbb{Z}^n$ mit $xAx^{tr} \leq C$ bis auf Vorzeichen.

Algorithmus 2.3.4. (POHST, ZASSENHAUS, 1989)

1. Initialisiere $M = \emptyset$.
2. Setze $q_{ij} \leftarrow a_{ij}$.

3. Für $i = 1, \dots, n$ setze $q_{ji} \leftarrow q_{ij}$, $q_{ij} \leftarrow (q_{ij}/q_{ii})$ ($i + 1 \leq j \leq n$) und anschließend für jedes i und $\mu = i + 1, \dots, n$: $q_{\mu\nu} \leftarrow q_{\mu\nu} - q_{\mu i}q_{i\nu}$.
4. Setze $q_{ij} \leftarrow 0$ für $1 \leq j < i \leq n$. Dann gilt:
$$xAx^{tr} = \sum_{i=1}^n q_{ii}(x_i + \sum_{j=i+1}^n q_{ij}x_j)^2.$$
5. Setze $i \leftarrow n$, $T_i \leftarrow C$, $U_i \leftarrow 0$.
6. Setze $Z \leftarrow \sqrt{T_i/q_{ii}}$, $UB(x_i) \leftarrow \lfloor Z - U_i \rfloor$, $x_i \leftarrow \lfloor -Z - U_i \rfloor - 1$.
7. Setze $x_i \leftarrow x_i + 1$. Falls $x_i \leq UB(x_i)$, gehe zu Schritt 9.
8. Setze $i \leftarrow i + 1$, gehe zu Schritt 7.
9. Falls $i = 1$, gehe zu Schritt 10, sonst setze $i \leftarrow i - 1$, $U_i \leftarrow \sum_{j=i+1}^n q_{ij}x_j$, $T_i \leftarrow T_{i+1} - q_{i+1,i+1}(x_{i+1} + U_{i+1})^2$ und gehe zu Schritt 6
10. Falls $x = 0$: gib M aus; Sonst $M \leftarrow M \cup \{x\}$ und gehe zu Schritt 8

Wie in Satz 2.1.10 gesehen, ist für positiv definite Gitter die Automorphismengruppe stets endlich. PLESKEN und SOUVIGNIER haben in [PS97] einen Algorithmus angegeben, mit dem ein Erzeugendensystem der Automorphismengruppe berechnet werden kann.

Die Automorphismen werden dazu zeilenweise konstruiert. Sei dazu A die GRAM-Matrix eines Gitters L mit Skalarprodukt Φ bezüglich einer Basis $B = (b_1, \dots, b_n)$. Ist nun

$$m := \max\{A_{ii} \mid 1 \leq i \leq n\}$$

der größte Diagonaleintrag von A , so setze $M := \{v \in L \mid \Phi(v, v) \leq m\}$. Für $k \leq n$ bildet ein Tupel $(v_1, \dots, v_k) \in L^k$ einen k -partiellen Automorphismus von L , falls $\Phi(v_i, v_j) = A_{ij}$ für alle $i, j \leq k$, die n -partiellen Automorphismen sind demnach Automorphismen von L . Dass jeder Automorphismus von L auch n -partiell ist, folgt aus folgendem Satz:

Satz 2.3.5. (WITTScher *Fortsetzungssatz*, 1937) Sei \mathcal{V} ein endlich dimensionaler Vektorraum über einem Körper K mit einer nicht ausgearteten symmetrischen oder schiefsymmetrischen Bilinearform Φ . Weiter sei

$$\varphi : \mathcal{U} \rightarrow \mathcal{U}'$$

eine Isometrie von Teilräumen bezüglich Φ . Dann setzt sich φ zu einem Automorphismus von \mathcal{V} fort.

BEWEIS. [Kne02, Satz 3.4 und Satz 4.3]

□

Satz 2.3.5 bleibt richtig, wenn man statt Vektorräumen über Körpern freie Moduln über lokalen Ringen zulässt. Allerdings ist er für Gitter im Allgemeinen falsch, so dass es keine Garantie gibt, dass sich ein k -partieller Automorphismus zu einem $(k + 1)$ -partiellen Automorphismus von L fortsetzt. Daher ist es wünschenswert, diejenigen partiellen Automorphismen möglichst früh zu erkennen, die sich nicht fortsetzen lassen.

In [PS97] werden hierzu mehrere Methoden diskutiert, für Details sei auf die Originalarbeit verwiesen.

2.4 Der VORONOI-Algorithmus

In diesem Abschnitt wird eine kurze Beschreibung des klassischen VORONOI-Algorithmus gegeben, die im Wesentlichen auf den Kapiteln 3 und 7 von [Mar03] aufbaut. Mit Hilfe dieses Verfahrens können in gegebener Dimension bis auf Reskalierung alle perfekten quadratischen Formen bzw. perfekten Gitter bestimmt werden. Alle Gitter sind auch in diesem Abschnitt, soweit nicht explizit anders gesagt, als positiv definit vorausgesetzt.

Definition 2.4.1. *Sei L ein positiv definites Gitter mit Skalarprodukt Φ . Weiter sei $\mathcal{V} := \mathbb{R} \otimes_{\mathbb{Z}} L$ und $\dim \mathcal{V} = n$.*

1. Zu $x \in \mathcal{V}$ heißt

$$\varphi_x : \text{End}_{\text{sym}}(\mathcal{V}) \rightarrow \mathbb{R}, \lambda \mapsto \Phi(\lambda(x), x)$$

die von x induzierte Linearform. Hierbei bezeichnet $\text{End}_{\text{sym}}(\mathcal{V})$ den Raum der symmetrischen Endomorphismen von \mathcal{V} .

2. Das Gitter L heißt *perfekt*, falls der Dualraum $\text{End}_{\text{sym}}(\mathcal{V})^*$ von den induzierten Linearformen der kürzesten Vektoren von L erzeugt wird,

$$\text{End}_{\text{sym}}(\mathcal{V})^* = \langle \varphi_x \mid x \in M(L) \rangle_{\mathbb{R}}.$$

3. Man nennt L *eutaktisch*, wenn es zu jedem $x \in M(L)$ ein $\rho_x > 0$ gibt, so dass

$$\text{Spur} = \sum_{x \in M(L)} \rho_x \varphi_x$$

erfüllt ist. Die ρ_x heißen Eutaxiekoeffizienten von L .

4. Das Gitter L heißt *extrem*, falls die HERMITE-Konstante

$$\gamma(L) := \frac{\min L}{\sqrt[n]{|\Delta(L)|^2}}$$

in L ein lokales Maximum annimmt.

Analoge Eigenschaften gelten für positiv definite Matrizen, indem man φ_x durch $P_x := x^{\text{tr}} x$ und $\text{End}_{\text{sym}}(\mathcal{V})$ durch $\mathbb{R}_{\text{sym}}^{n \times n}$ ersetzt. Man nennt dann die induzierte quadratische Form entsprechend *perfekt*, *eutaktisch* bzw. *extrem*. Man nennt $\text{End}_{\text{sym}}(\mathcal{V})$ bzw. $\mathbb{R}_{\text{sym}}^{n \times n}$ auch den VORONOI-Raum. .

Satz 2.4.2. (KORKIN und SOLOTARJOW, 1873) *Ein perfektes Gitter mit gegebener Norm m ist bis auf Isometrie durch die Menge seiner kürzesten Vektoren eindeutig bestimmt.*

BEWEIS. [Mar03, Theorem 3.2.10]. \square

Extreme Gitter sind v.a. im Zusammenhang mit dichtesten Kugelpackungen interessant. In [Vor08] gibt VORONOI folgende Charakterisierung extremer Gitter:

Satz 2.4.3. (VORONOI, 1908)

Ein Gitter ist dann und nur dann extrem, wenn es sowohl perfekt als auch eutaktisch ist.

BEWEIS. [Mar03, Theorem 3.4.6]. \square

Die Tatsache, dass man der Klassifikation perfekter Gitter algorithmisch beikommen kann, folgt aus einem weiteren Resultat von VORONOI formuliert in der Sprache der Matrizen.

Satz 2.4.4. (VORONOI, 1908)

Sei $A \in \mathbb{R}_{>0}^{n \times n}$ eine positiv definite Matrix und sei $K > 0$. Dann ist die Menge

$$M_K := \{B \in \mathbb{R}_{\geq 0}^{n \times n} \mid B \text{ perfekt mit } \min B = m \text{ und } \text{Spur}(A \cdot B) \leq K\}$$

endlich. Hierbei bezeichnen $\mathbb{R}_{>0}^{n \times n}$ bzw. $\mathbb{R}_{\geq 0}^{n \times n}$ die Mengen der positiv (semi-)definiten reellen Matrizen. $\mathbb{R}_{\geq 0}^{n \times n}$ nennt man auch den VORONOI-Kegel.

BEWEIS. [Mar03, Theorem 7.3.2] \square

Der zentrale Begriff für den VORONOI-Algorithmus ist nun der folgende:

Definition 2.4.5. *Unter dem VORONOI-Bereich $\mathcal{D}(L)$ eines Gitters L (einer symmetrischen Matrix A) versteht man die von seinen (ihren) kürzesten Vektoren aufgespannte Pyramide,*

$$\mathcal{D}(L) = \left\{ \sum_{x \in M(L)} \lambda_x \pi_x \mid \lambda_x \geq 0 \right\}$$

bzw.

$$\mathcal{D}(A) = \left\{ \sum_{x \in M(L)} \lambda_x P_x \mid \lambda_x \geq 0 \right\}.$$

Hierbei bezeichnet π_x die Projektion auf $\langle x \rangle_{\mathbb{R}}$ und P_x wie oben die Matrix xx^{tr} .

Man erhält unmittelbar:

Proposition 2.4.6. *Sei L ein Gitter (bzw. A eine positiv definite Matrix). Dann sind folgende Aussagen äquivalent:*

1. *L ist perfekt (bzw. A ist perfekt).*
2. *Der VORONOI-Bereich von L (bzw. A) ist in keiner Hyperebene des VORONOI-Raumes enthalten.*

3. Das Innere des VORONOI-Bereiches ist nicht leer.

Zur Vereinfachung der Notation betrachten wir ab jetzt nur noch positiv definite Matrizen anstelle von Gittern. Für symmetrische Matrizen A, B sei ab jetzt

$$(A, B) := \text{Spur}(A \cdot B).$$

Definition 2.4.7. Sei A eine perfekte Matrix und $\mathcal{D}(A)$ ihr VORONOI-Bereich. Eine Matrix $R \in \mathbb{R}_{sym}^{n \times n}$ heißt eine Richtung von A , falls

$$\begin{aligned} \text{Spur}(P_x \cdot R) = xRx^{tr} &\geq 0 \text{ für alle } x \in M(A) \\ xRx^{tr} &= 0 \text{ für } n-1 \text{ linear unabhängige } x \in M(A). \end{aligned}$$

Die Menge

$$\mathcal{F}(R) := \{B \in \mathbb{R}_{sym}^{n \times n} \mid (R, B) = 0\} \cap \mathcal{D}(A)$$

nennt man die zu R gehörige Facette von $\mathcal{D}(A)$.

Satz 2.4.8. Sei A eine perfekte Matrix mit Minimum m und sei \mathcal{F} eine Facette des VORONOI-Bereiches $\mathcal{D}(A)$. Weiterhin sei R eine Richtung von A . Definiere $A_\theta = A + \theta R$, $\theta \in \mathbb{R}$.

- (i) Es gibt genau ein $\rho > 0$, so dass A_θ nicht perfekt mit Minimum m ist für jedes $0 < \theta < \rho$ und für $\theta > \rho$ ein echt kleineres Minimum hat oder nicht positiv definit ist.
- (ii) Für $0 < \theta < \rho$ ist $M(A_\theta) = \{x \in M(A) \mid P_x \in \mathcal{F}\}$.
- (iii) Für $\theta < 0$ ist $m(A_\theta) < m$ oder A_θ ist nicht positiv definit.
- (iv) Die Form A_ρ ist perfekt mit Minimum m , von A verschieden und zusammen mit A die einzige perfekte Form mit Minimum m , deren VORONOI-Bereich die Facette \mathcal{F} enthält.

BEWEIS. [Mar03, Theorem 7.2.1] □

Definition 2.4.9. Die Form A_ρ aus Satz 2.4.8 heißt zu A benachbart oder auch ein Nachbar von A bezüglich \mathcal{F} (in Richtung R).

Definition 2.4.10. Der Graph Γ , dessen Knotenmenge $V(\Gamma)$ die Menge aller perfekten Matrizen mit Minimum m sei, und dessen Kantenmenge gegeben ist durch

$$E(\Gamma) = \{(A, A') \in V(\Gamma) \times V(\Gamma) \mid A, A' \text{ benachbart}\}$$

heißt der (n -dimensionale) VORONOI-Graph.

Wählt man als Knotenmenge die Menge der Äquivalenzklassen perfekter Matrizen mit Minimum m , so nennt man den resultierenden Graphen Γ_C den VORONOI-Klassengraphen. Hierbei heißen zwei positiv definite Matrizen A, B äquivalent, falls eine Matrix $T \in \text{GL}_n(\mathbb{Z})$ existiert mit $TAT^{tr} = B$.

Satz 2.4.11. Sei A eine perfekte Matrix und A' positiv definit. Dann existieren perfekte Matrizen $A = A_0, A_1, \dots, A_k$, so dass $A' \in \mathcal{D}_{A_k}$ und A_i zu A_{i+1} benachbart ist für $0 \leq i < k$.

BEWEIS. [Mar03, Theorem 7.4.1] □

Satz 2.4.12. *Seien Q und Q' perfekte Formen. Dann existieren endlich viele perfekte Formen $Q = Q_0, Q_1, \dots, Q_k = Q'$, so dass Q_i zu Q_{i+1} benachbart ist für $0 \leq i < k$.*

BEWEIS. [Mar03, Theorem 7.4.3] □

Satz 2.4.12 liefert damit sofort das

Korollar 2.4.13. *Der VORONOI-Graph ist zusammenhängend und der VORONOI-Klassengraph ist endlich.*

Damit haben wir die nötigen Begriffe, um den VORONOI-Algorithmus zu formulieren.

Algorithmus 2.4.14. (VORONOI, 1908)

1. Wähle eine perfekte Matrix A_0 .
2. Bestimme einen Repräsentanten zu jeder $\text{Aut}(A_0)$ -Bahn der Facetten von \mathcal{D}_{A_0} .
3. Bestimme für jeden Repräsentanten aus 2 die benachbarte Form.
4. Falls eine so neu gefundene Form äquivalent zu einer bereits gefundenen Form ist, entferne diese Form.
5. Wiederhole Schritte 2 bis 4.
6. Sobald jede neu konstruierte Form äquivalent zu einer bereits gefundenen Form ist, STOP.

Da der VORONOI-Klassengraph endlich ist, terminiert dieser Algorithmus. Um ihn konkret auszuführen, muss man offenbar algorithmisch in der Lage sein,

1. zu einer gegebenen Form A die $\text{Aut}(A)$ -Bahnen der Facetten ihres VORONOI-Bereiches zu bestimmen,
2. zu einer Form A und einem Facettenvektor R die Zahl $\rho > 0$ zu bestimmen, so dass $A + \rho R$ zu A in Richtung R benachbart ist,
3. Äquivalenz quadratischer Formen zu entscheiden.

Das Problem 1 ist das Wesentliche für diesen Algorithmus. Seine Komplexität hängt im Wesentlichen sowohl quadratisch von der Dimension n als auch von der Differenz $\kappa(A) - \frac{n(n+1)}{2}$ ab. Die Existenz des Gitters

$$\mathbb{E}_8 := \mathbb{D}_8 \cup \left(\frac{1}{2} \sum_{i=1}^8 e_i + \mathbb{D}_8 \right)$$

mit Kusszahl $\kappa(\mathbb{E}_8) = 240 \gg 36 = \frac{8 \cdot (8+1)}{2}$ (vgl. [Mar03, Theorem 4.4.4]) macht es praktisch unmöglich, den VORONOI-Algorithmus in Dimensionen ≥ 8 ohne Weitere Überlegungen explizit auszuführen.

Das Problem 3 ist zwar an sich von Interesse, aber für den VORONOI-Algorithmus nicht essentiell, da er ohnehin nur bis Dimension 7 durchführbar ist.

Für Problem 2 sei auf [Mar03, Section 7.8] bzw. Proposition 3.1.11 verwiesen.

Kapitel 3

Duale Kegel

In diesem Kapitel werden wir einige Aspekte und Ideen aus Abschnitt 2.4 wiederaufnehmen, um so das theoretische Gerüst für die angesprochenen Algorithmen bereitzustellen. Die ersten beiden Abschnitte basieren hauptsächlich auf [Opg01], [Ser80] und [Dic80]. Im dritten Abschnitt wird ein Fundamentalbereich für die Operation von Ω^{ad} auf $\mathcal{V}_2^{>0}$ konstruiert so wie einige Kriterien für die Endlichkeit des Restklassengraphen Γ_D/Ω bewiesen.

3.1 Kegel und perfekte Punkte

Für diesen Abschnitt seien \mathcal{V}_1 und \mathcal{V}_2 reelle Vektorräume der Dimension n . Weiter sei

$$\sigma : \mathcal{V}_1 \times \mathcal{V}_2 \rightarrow \mathbb{R}$$

eine in beiden Komponenten lineare und nicht ausgeartete Abbildung, d.h. für $v_1 \in \mathcal{V}_1$ mit $\sigma(v_1, w) = 0$ für alle $w \in \mathcal{V}_2$ folgt $v_1 = 0$ und umgekehrt.

Definition 3.1.1. *Zwei Teilmengen $\mathcal{V}_1^{>0} \subset \mathcal{V}_1$ und $\mathcal{V}_2^{>0} \subset \mathcal{V}_2$ heißen duale Kegel bezüglich σ , wenn sie folgende Eigenschaften erfüllen:*

- (DC1) $\mathcal{V}_i^{>0}$ ist offen in \mathcal{V}_i und nicht leer für $i = 1, 2$.
- (DC2) Für alle $v_i \in \mathcal{V}_i^{>0}$ ($i = 1, 2$) gilt $\sigma(v_1, v_2) > 0$.
- (DC3) Für jedes $v_1 \in \mathcal{V}_1 \setminus \mathcal{V}_1^{>0}$ gibt es ein $v_2 \in \mathcal{V}_2^{>0} \setminus \{0\}$ mit $\sigma(v_1, v_2) \leq 0$. Ebenso existiert für jedes $v_2 \in \mathcal{V}_2 \setminus \mathcal{V}_2^{>0}$ ein $v_1 \in \mathcal{V}_1^{>0} \setminus \{0\}$ mit $\sigma(v_1, v_2) \leq 0$. Hierbei bezeichnet $\mathcal{V}_i^{\geq 0}$ den Abschluss von $\mathcal{V}_i^{>0}$ in \mathcal{V}_i für $i = 1, 2$. Der Rand von $\mathcal{V}_i^{\geq 0}$ wird mit $\partial\mathcal{V}_i^{\geq 0}$ bezeichnet.

Für den Rest des Abschnittes seien nun $\mathcal{V}_1^{>0}$ und $\mathcal{V}_2^{>0}$ feste duale Kegel in \mathcal{V}_1 und \mathcal{V}_2 bezüglich der Abbildung σ . Im folgenden Lemma finden sich einige elementare Eigenschaften zusammen. Ursprünglich finden sich die Aussagen bereits in [Koe57].

Lemma 3.1.2. (i) Für $x, y \in \mathcal{V}_1^{>0}$ und positive Zahlen $a, b > 0$ ist auch $ax + by \in \mathcal{V}_1$.

(ii) Für $v_1 \in \mathcal{V}_1^{\geq 0} \setminus \{0\}$ und $v_2 \in \mathcal{V}_2^{>0}$ gilt $\sigma(v_1, v_2) > 0$.

- (iii) Für jedes $v_1 \in \mathcal{V}_1 \setminus \mathcal{V}_1^{\geq 0}$ existiert ein $v_2 \in \mathcal{V}_2^{\geq 0}$ mit $\sigma(v_1, v_2) < 0$.
- (iv) Für jedes $v_1 \in \partial\mathcal{V}_1^{\geq 0}$ gibt es ein $v_2 \in \partial\mathcal{V}_2^{\geq 0}$ mit $\sigma(v_1, v_2) = 0$.
- (v) Gilt $x \in \mathcal{V}_1^{\geq 0}$ und $-x \in \mathcal{V}_1^{\geq 0}$, so ist $x = 0$.
- (vi) Sei Φ_2 ein positiv definites Skalarprodukt auf \mathcal{V}_2 und sei für $v_2 \in \mathcal{V}_2$ die Norm von v_2 definiert als $\|v_2\|_2 := \sqrt{\Phi_2(v_2, v_2)}$. Dann existiert zu jeder kompakten Teilmenge $A \subset \mathcal{V}_1$ ein $\rho(A) > 0$, so dass für alle $a \in A$ und $v_2 \in \mathcal{V}_2^{\geq 0}$ gilt, dass $\sigma(a, v_2) \geq \rho(A)\|v_2\|_2$.

BEWEIS. (i) Angenommen, die Behauptung ist falsch. Seien also $x, y \in \mathcal{V}_1^{\geq 0}$ und $a, b > 0$, so dass $ax + by \notin \mathcal{V}_1$. Nach (DC3) gibt es dann ein $z \in \mathcal{V}_2^{\geq 0} \setminus \{0\}$ mit $0 \geq \sigma(ax + by, z) = a\sigma(x, z) + b\sigma(y, z)$. Dies ist aber ein Widerspruch, da $a, b, \sigma(x, z)$ und $\sigma(y, z)$ positive Zahlen sind.

(ii) Sei $v_1 \in \mathcal{V}_1^{\geq 0} \setminus \{0\}$. Da σ stetig ist, ist $\sigma(v_1, v_2) \geq 0$ für alle $v_2 \in \mathcal{V}_2^{\geq 0}$. Angenommen, es gibt ein $v_2 \in \mathcal{V}_2$, so dass $\sigma(v_1, v_2) = 0$. Damit ist $v_1 \notin \mathcal{V}_1^{\geq 0}$. Da $\mathcal{V}_2^{\geq 0}$ nach (DC1) offen ist, gibt es zu jedem $w_2 \in \mathcal{V}_2$ ein $\varepsilon > 0$, so dass $v_2 + \varepsilon w_2 \in \mathcal{V}_2^{\geq 0}$. Daraus folgt

$$0 \leq \sigma(v_1, v_2 + \varepsilon w_2) = \sigma(v_1, v_2) + \varepsilon\sigma(v_1, w_2) = \varepsilon\sigma(v_1, w_2),$$

und damit $\sigma(v_1, w_2) \geq 0$ für alle $w_2 \in \mathcal{V}_2$. Da σ nicht ausgeartet ist, kann das aber nicht sein.

(iii) Angenommen, es gibt ein $v_1 \in \mathcal{V}_1 \setminus \mathcal{V}_1^{\geq 0}$, so dass $\sigma(v_1, v_2) \geq 0$ für alle $v_2 \in \mathcal{V}_2^{\geq 0}$. Nun ist $\mathcal{V}_1 \setminus \mathcal{V}_1^{\geq 0}$ offen, also gibt es zu jedem $w_1 \in \mathcal{V}_1^{\geq 0}$ ein $\varepsilon > 0$, so dass $v_1 + \varepsilon w_1 \notin \mathcal{V}_1^{\geq 0}$. (DC3) liefert dann die Existenz eines $w_2 \in \mathcal{V}_2^{\geq 0} \setminus \{0\}$ mit $\sigma(v_1 + \varepsilon w_1, w_2) \leq 0$, also

$$0 \geq \sigma(v_1 + \varepsilon w_1, w_2) = \sigma(v_1, w_2) + \varepsilon\sigma(w_1, w_2),$$

was aber ein Widerspruch ist, denn nach Annahme ist $\sigma(v_1, w_2) \geq 0$ und sowohl ε als auch $\sigma(w_1, w_2)$ sind echt positiv.

(iv) Analog zu (ii) und (iii).

(v) Nach (ii) ist $\sigma(v_1, v_2) \geq 0$ und $\sigma(-v_1, v_2) \geq 0$ für alle $v_2 \in \mathcal{V}_2$ und somit $\sigma(v_1, v_2) = 0$ für alle $v_2 \in \mathcal{V}_2$. Aufgrund der Stetigkeit von σ und der Tatsache, dass σ nicht ausgeartet ist, folgt die Behauptung, da $\mathcal{V}_2^{\geq 0}$ offen in \mathcal{V}_2 ist.

(vi) Aus Homogenitätsgründen ist es offenbar ausreichend, die Behauptung für $\|v_2\|_2 = 1$ zu zeigen. Die Menge der Tupel (a, v_2) mit $a \in A$ und $v_2 \in \mathcal{V}_2^{\geq 0}$ und $\|v_2\|_2 = 1$ bildet eine kompakte Teilmenge von $\mathcal{V}_1 \times \mathcal{V}_2$, auf der die stetige Abbildung σ ihr Minimum annimmt. Dieses Minimum ist nach (ii) strikt positiv, also folgt die Behauptung. \square

Bemerkung 3.1.3. Ist $D \subset \mathcal{V}_2^{\geq 0}$ diskret und $v_1 \in \mathcal{V}_1^{\geq 0}$ fest. Nach Lemma 3.1.2(vi) ist dann $\sigma(v_1, d) \geq \rho(v_1)\|d\|_2$ für jedes $d \in D$. Also ist insbesondere $\sigma(v_1, d) > 0$ für jedes $d \in D$ und die Menge $\{d \in D \mid \sigma(v_1, d) \leq c\}$ ist für jedes $c > 0$ endlich.

Damit ergibt folgende Definition Sinn.

Definition 3.1.4. Sei $D \subset \mathcal{V}_2^{\geq 0} \setminus \{0\}$ diskret und $v_1 \in \mathcal{V}_1^{\geq 0}$ fest.

- (i) Unter dem D -Minimum von v_1 versteht man $\mu_D(v_1) := \min\{\sigma(v_1, d) \mid d \in D\}$.
- (ii) Die Menge $M_D(v_1) := \{d \in D \mid \sigma(v_1, d) = \mu_D(v_1)\}$ heißt die Menge der D -minimalen Vektoren von v_1 .
- (iii) $\mathcal{D}_D(v_1) := \{\sum_{d \in M_D(v_1)} \lambda_d d \mid \lambda_d \geq 0\}$ heißt der D -VORONOI-Bereich von v_1 .
- (iv) Ein Vektor $v_1 \in \mathcal{V}_1^{>0}$ heißt D -perfekt, falls sein D -VORONOI-Bereich ein nicht-leeres Inneres hat, also $\mathcal{D}_D(v_1)^\circ \neq \emptyset$. Mit P_D sei die Menge aller D -perfekten Vektoren mit D -Minimum 1 bezeichnet.

Bemerkung 3.1.5. In der Situation von Definition 3.1.4 gilt

- (i) Für $\lambda > 0$ ist $M_D(v_1) = M_D(\lambda v_1)$.
- (ii) Ein Vektor $v_1 \in \mathcal{V}_1^{>0}$ ist genau dann D -perfekt, wenn $\dim\langle M_D(v_1) \rangle = n$.
- (iii) Notwendig, aber sicherlich nicht hinreichend dafür, dass $P_D \neq \emptyset$ gilt, ist offenbar, dass D eine Basis von \mathcal{V}_2 enthält.

Lemma 3.1.6. Seien $\mathcal{V}_1^{>0}$ und $\mathcal{V}_2^{>0}$ duale Kegel bezüglich σ und sei $D \subset \mathcal{V}_2^{>0} \setminus \{0\}$ diskret in \mathcal{V}_2 .

- (i) Für jedes $x \in \mathcal{V}_1^{>0}$ gibt es eine offene Umgebung $U \subset \mathcal{V}_1^{>0}$ von x , so dass $M_D(u) \subseteq M_D(x)$ für alle $u \in U$.
- (ii) Die Funktion μ_D ist stetig.

BEWEIS. Sei $r > 0$, so dass $B := \overline{B_r(x)} \subset \mathcal{V}_1^{>0}$, wobei $B_r(x)$ die offene Kugel um x mit Radius r bezeichne. Da B kompakt ist gibt es nach Lemma 3.1.2vi ein $\rho > 0$ mit $\sigma(b, d) \geq \rho \|d\|_2$ für alle $b \in B$ und alle $d \in D$.

Sei nun $\varepsilon > 0$. Da D diskret ist, ist die Menge

$$S := \{d \in D \mid \sigma(b, d) \leq \mu_D(x) + \varepsilon \text{ für ein } b \in B\}$$

endlich und es gilt $M_D(x) \subset S$.

Es existiert eine reelle Zahl η mit

$$0 < \eta < \min(\{\varepsilon\} \cup \{\sigma(x, d) - \mu_D(x) \mid d \in S \setminus M_D(x)\}).$$

Da $\sigma(\cdot, d)$ für jedes feste $d \in D$ stetig und M endlich ist, gibt es also ein $\delta > 0$ mit $|\sigma(y, d) - \sigma(x, d)| < \frac{\eta}{2}$ für alle $d \in S$ und alle $y \in B_\delta(0)$. Ohne Einschränkung kann man zusätzlich $\delta < r$ annehmen, so dass $x + y \in B$ für $y \in B_\delta(0)$. Für diese y gilt dann

$$\sigma(x + y, d) = \sigma(x, d) + \sigma(y, d) \begin{cases} < \mu_D(x) + \frac{\eta}{2} & , \text{ falls } d \in M_D(x) \\ > \mu_D(x) + \frac{\eta}{2} & , \text{ falls } d \in S \setminus M_D(x) , \\ > \mu_D(x) + \frac{\eta}{2} & , \text{ falls } d \in D \setminus S \end{cases}$$

was $\sigma(x + y, d) > \mu_D(x) + \varepsilon > \mu_D(x) + \frac{\eta}{2}$ für alle $d \in D \setminus S$ und $x + y \in B$ impliziert. Damit ist $M_D(x + y) \subseteq M_D(x)$ für $y \in B_\delta(0)$, also ist (i) gezeigt.

Für $y \in B_\delta(0)$ mit δ wie oben gilt nach Konstruktion $|\mu_D(x) - \mu_D(x + y)| < \frac{\eta}{2}$. Da η im Wesentlichen beliebig gewählt war, folgt so auch die Stetigkeit von μ_D , also (ii). \square

Wir beschränken uns im Folgenden auf diskrete Mengen D mit folgender Eigenschaft:

Definition 3.1.7. *Man nennt eine diskrete Menge $D \subset \mathcal{V}_2^{\geq 0} \setminus \{0\}$ zulässig, falls für jede Folge $(x_k)_{k \in \mathbb{N}}$ in $\mathcal{V}_1^{\geq 0}$ mit Grenzwert $x \in \partial\mathcal{V}_1^{\geq 0}$ die Folge $(\mu_D(x_k))_{k \in \mathbb{N}}$ gegen 0 konvergiert.*

Lemma 3.1.8. *Eine diskrete Menge $D \subset \mathcal{V}_2^{\geq 0} \setminus \{0\}$ ist genau dann zulässig, wenn es für jedes $x \in \partial\mathcal{V}_1^{\geq 0}$ und $\varepsilon > 0$ ein $d \in D$ gibt mit $\sigma(x, d) < \varepsilon$.*

BEWEIS. Sei zunächst $D \subset \mathcal{V}_2^{\geq 0} \setminus \{0\}$ zulässig, $x \in \partial\mathcal{V}_1^{\geq 0}$ beliebig und $(x_k)_{k \in \mathbb{N}}$ eine Nullfolge mit $x_k \in \mathcal{V}_1^{\geq 0}$ für alle $k \in \mathbb{N}$. Dann ist $x + x_k \in \mathcal{V}_1^{\geq 0}$ für alle $k \in \mathbb{N}$ und für jedes $k \in \mathbb{N}$ existiert ein $d_k \in D$ mit

$$\mu_D(x + x_k) = \sigma(x + x_k, d_k) = \sigma(x, d_k) + \sigma(x_k, d_k) \geq \sigma(x, d_k) \geq 0$$

Nach Voraussetzung gilt $\mu_D(x + x_k) \rightarrow 0$, $k \rightarrow \infty$, also folgt die Behauptung.

Sei nun umgekehrt $x \in \partial\mathcal{V}_1^{\geq 0}$ beliebig und $(x_k)_{k \in \mathbb{N}}$ eine Folge in $\mathcal{V}_1^{\geq 0}$ mit Grenzwert x . Zu $\varepsilon > 0$ gibt es dann nach Voraussetzung ein $d \in D$ mit $\sigma(x, d) < \frac{\varepsilon}{2}$, außerdem gibt es ein $N \in \mathbb{N}$ mit der Eigenschaft, dass

$$|\sigma(x_k - x, d)| < \frac{\varepsilon}{2} \text{ für alle } k \geq N$$

Es folgt

$$\mu_D(x_k) \leq |\sigma(x_k, d)| \leq |\sigma(x, d)| + |\sigma(x_k - x, d)| < \varepsilon$$

für $k \geq N$. Damit ist D zulässig. \square

Lemma 3.1.9. *Sei $D \subset \mathcal{V}_2^{\geq 0} \setminus \{0\}$ diskret und zulässig. Dann ist auch P_D eine diskrete Menge.*

BEWEIS. Sei $(x_k)_{k \in \mathbb{N}}$ eine Folge in P_D , die gegen $x \in \mathcal{V}_1^{\geq 0}$ konvergiert. Angenommen, es wäre $x \in \partial\mathcal{V}_1^{\geq 0}$. Dann würde $\mu_D(x_k) \rightarrow 0$ folgen, da D zulässig ist, was aber nicht sein kann, da nach Definition von P_D gilt, dass $\mu_D(x_k) = 1$ für alle $k \in \mathbb{N}$.

Demnach gilt $x \in \mathcal{V}_1^{\geq 0}$. Nach Lemma 3.1.6 gibt es also eine Umgebung U von x in $\mathcal{V}_1^{\geq 0}$ mit $M_D(u) \subseteq M_D(x)$ für alle $u \in U$. Insbesondere gibt es daher ein $N \in \mathbb{N}$, so dass $M_D(x_k) \subseteq M_D(x)$ für alle $k \geq N$. Sei also $k \geq N$. Da x_k perfekt ist, gibt es linear unabhängige Vektoren $d_1, \dots, d_n \in M_D(x_k) \subseteq M_D(x)$. Nun gilt aber für $1 \leq j \leq n$, dass

$$\sigma(x_k - x/\mu_D(x), d_j) = 0$$

und da σ nicht ausgeartet ist, muss daher $x_k - x/\mu_D(x) = 0$ für $k \geq N$ gelten. Damit ist jede konvergente Folge in P_D schließlich konstant, demnach ist P_D diskret. \square

Aus dem Beweis erhält man sofort in Analogie zu Satz 2.4.2:

Korollar 3.1.10. *Seien x, y D -perfekte Vektoren und $d_1, \dots, d_k \in M_D(x) \cap M_D(y)$ mit $k \geq n$ so, dass $\{d_1, \dots, d_k\}$ eine Basis von \mathcal{V}_2 enthält. Dann gibt es ein $\lambda > 0$ mit $x = \lambda y$. D -perfekte Vektoren sind also durch die Menge ihrer kürzesten Vektoren bis auf Vielfache eindeutig bestimmt.*

Der Beweis der folgenden Proposition gibt eine konstruktive Methode zur Berechnung D -perfekter Vektoren an.

Proposition 3.1.11. *Sei $D \subset \mathcal{V}_2^{\geq 0} \setminus \{0\}$ diskret in \mathcal{V}_2 und zulässig. Dann gibt es zu jedem $y \in \mathcal{V}_1^{\geq 0}$ ein $x \in P_D$, so dass $M_D(y) \subseteq M_D(x)$.*

BEWEIS. Es sei $k := \dim\langle M_D(y) \rangle$. Ist $k = n$, so liefert $x = y/\mu_D(y)$ das Gewünschte.

Ist $k < n$, so gibt es ein $z \in \mathcal{V}_1 \setminus \{0\}$ mit $\sigma(z, d) = 0$ für alle $d \in M_D(y)$. Ohne Einschränkung sei $z \notin \mathcal{V}_1^{\geq 0}$, sonst ersetze man z durch $-z$. Nach der Wahl von z existiert nun ein $\lambda_0 > 0$ mit $y + \lambda_0 z \in \partial\mathcal{V}_1^{\geq 0}$. Nach Voraussetzung ist D zulässig, also gilt $\mu(y + \lambda z) \rightarrow 0$ für $\lambda \rightarrow \lambda_0$. Da μ_D stetig ist, existiert nach dem Zwischenwertsatz ein $\lambda_1 \in (0, \lambda_0)$ mit $\mu_D(y) > \mu_D(y + \lambda_1 z) > 0$. Sei $M := \{d \in D \mid \sigma(y + \lambda_1 z, d) \leq \mu_D(y)\}$. Nach Bemerkung 3.1.3 ist M endlich und offenbar eine echte Obermenge von $M_D(y)$. Für alle $d \in M \setminus M_D(y)$ gilt $\sigma(z, d) < 0$ und $\sigma(y, d) > \mu_D(y)$. Definiere

$$\lambda_2 := \min \left\{ \frac{\mu_D(y) - \sigma(y, d)}{\sigma(z, d)} \mid d \in M \setminus M_D(y) \right\},$$

$$y_2 := y + \lambda_2 z.$$

Man hat $0 < \lambda_2 < \lambda_1$ und $\sigma(y_2, d) > \mu_D(y)$ für $d \in D \setminus M$, denn $\sigma(y, d) > \mu_D(y)$ und $\sigma(y + \lambda_1 z, d) > \mu_D(y)$. Nach Konstruktion gilt $\sigma(y_2, d) \geq \mu_D(y)$ für alle $d \in M$ mit Gleichheit für $d \in M_D(y)$ und mindestens ein $d \in M \setminus M_D(y)$. Dieses d kann nicht im Erzeugnis von $M_D(y)$ liegen, da $z \in M_D(y)^\perp$, also wäre anderenfalls $\mu_D(y) = \sigma(y_2, d) = \sigma(y, d)$, also $d \in M_D(y)$. Es folgt daher $M_D(y) \subsetneq M_D(y_2)$ und $\dim\langle M_D(y_2) \rangle \geq k$. Iteration dieses Verfahrens liefert die Behauptung. \square

Ab hier betrachten wir den D -VORONOI-Bereich eines D -perfekten Vektors x .

Bemerkung 3.1.12. *Der D -VORONOI-Bereich $\mathcal{D}_D(x)$ ist per Definition die Menge aller Linearkombinationen der Vektoren aus $M_D(x)$ mit nichtnegativen Koeffizienten, bildet also eine Pyramide in $\mathcal{V}_2^{\geq 0}$ mit Spitze im Ursprung. Die dazu duale Beschreibung erhält man durch endlich viele lineare Ungleichungen der Form $\sigma(y, \cdot) \geq 0$ für gewisse $y \in \mathcal{V}_1$.*

Geometrisch benutzt die erste Beschreibung die begrenzenden Strahlen der Pyramide, die zweite die begrenzenden Hyperflächen.

Definition 3.1.13. *Eine Richtung für x ist ein Vektor $r \in \mathcal{V}_1 \setminus \{0\}$ mit $\sigma(r, z) \geq 0$ für alle $z \in M_D(x)$ und $\sigma(r, z) = 0$ für $n - 1$ linear unabhängige Vektoren $z \in M_D(x)$. Die zu einer Richtung r gehörige Facette $\mathcal{F}(r)$ von $\mathcal{D}_D(x)$ ist definiert durch*

$$\mathcal{F}(r) := \mathcal{D}_D(x) \cap \{z \in \mathcal{V}_2 \mid \sigma(r, z) = 0\}.$$

Bemerkung 3.1.14. *Liegt eine Richtung r von x in $\mathcal{V}_1^{\geq 0}$, so gilt*

(i) $\sigma(r, d) \geq 0$ für alle $d \in D$ und

(ii) $\sigma(r, z) = 0$ für alle $z \in \mathcal{F}(r)$.

Mit (i) folgt, dass für $\lambda > 0$ gilt: $M_D(x + \lambda r) = M_D(x) \cap W(r)$, also ist $x + \lambda r$ nicht D -perfekt. (ii) impliziert, dass $\mathcal{F}(r)$ im Rand von $\mathcal{V}_2^{\geq 0}$ enthalten ist und dass r ein Randpunkt von $\mathcal{V}_1^{\geq 0}$ ist. In diesem Fall nennt man die Richtung r blind.

Ganz ähnlich wie in Proposition 3.1.11 lassen sich auch aus bekannten D -perfekten Punkten weitere konstruieren:

Proposition 3.1.15. *Sei $x \in P_D$ für eine zulässige, diskrete Menge $D \subset \mathcal{V}_2^{\geq 0} \setminus \{0\}$. Ist $r \notin \mathcal{V}_1^{\geq 0}$ eine (nicht blinde) Richtung von x , so existiert ein $\rho > 0$, so dass $x + \rho r$ D -perfekt mit D -Minimum 1 ist.*

BEWEIS. Zu r gibt es nach Definition $n - 1$ linear unabhängige Vektoren $d_1, \dots, d_{n-1} \in M_D(x)$ mit $\sigma(r, d_i) = 0$ für $i = 1 \dots n - 1$. Diese erzeugen die Facette $\mathcal{F}(r)$ von r als $n - 1$ -dimensionalen Kegel.

Da r nach Voraussetzung nicht im Kegel $\mathcal{V}_1^{\geq 0}$ liegt, gibt es ein $\lambda_0 > 0$, so dass $x + \lambda_0 r \in \partial \mathcal{V}_1^{\geq 0}$ liegt. Mit derselben Argumentation wie im Beweis zu Proposition 3.1.11 kann man also, da D zulässig ist, ein $\lambda_0 > \rho > 0$ mit der Eigenschaft, dass $d_i \in M_D(x + \rho r)$ für $i = 1 \dots n - 1$ und $x + \rho r$ perfekt ist, konstruieren. \square

Definition 3.1.16. *Der Vektor $x + \rho r$ wie am Schluss der letzten Proposition heißt Nachbar von x in Richtung r .*

Bemerkung 3.1.17. *Neben der Methode im Beweis von Proposition 3.1.11 lässt sich ein Nachbar eines Punktes x in einer Richtung r auch durch folgendes Verfahren bestimmen:*

Sei $y = x + \lambda_1 r \in \mathcal{V}_1^{\geq 0}$ so, dass $\mu_D(y) < \mu_D(x)$ und sei $y_1 \in M_D(y)$. Dann gilt folgende Abschätzung:

$$\mu_D(x) > \sigma(y, y_1) = \underbrace{\sigma(x, y_1)}_{> \mu_D(x)} + \lambda_1 \sigma(r, y_1),$$

also muss $\sigma(r, y_1) < 0$ gelten. Damit ist dann die Zahl

$$\tilde{\rho} = \frac{\mu_D(x) - \sigma(x, y_1)}{\sigma(r, y_1)}$$

echt positiv. Es gilt nach Konstruktion

$$\sigma(x + \tilde{\rho} r, y_1) = \mu_D(x).$$

Ersetzt man also y durch $x + \tilde{\rho} r$ und ist $\mu_D(y) = \mu_D(x)$, so ist $\tilde{\rho} = \rho$ wie in Proposition 3.1.15 und der Nachbar ist gefunden. Falls nicht, wiederhole obigen Vorgang.

Der nun folgende Satz, das Hauptresultat dieses Abschnittes, ist eine Anpassung und Verallgemeinerung von Satz 2.4.12 auf die hier gegebene Situation:

Satz 3.1.18. *Ist $D \subset \mathcal{V}_2^{\geq 0} \setminus \{0\}$ diskret in \mathcal{V}_2 und zulässig, so bilden die D -VORONOI-Bereiche der D -perfekten Vektoren eine exakte Kachelung von $\mathcal{V}_2^{\geq 0}$, d.h. dass jede Facette eines D -VORONOI-Bereiches eine Facette von genau zwei D -VORONOI-Bereichen ist und die D -VORONOI-Bereiche den gesamten Kegel $\mathcal{V}_2^{\geq 0}$ überdecken.*

BEWEIS. Seien $x_1 \neq x_2 \in P_D$. Zu zeigen ist zunächst, dass $\mathcal{D}_D(x_1)^\circ \cap \mathcal{D}_D(x_2)^\circ = \emptyset$.

Angenommen, das wäre falsch und es gäbe ein $x \in \mathcal{D}_D(x_1)^\circ \cap \mathcal{D}_D(x_2)^\circ$. Ohne Einschränkung kann man $\sigma(x_1, x) = 1$ und $\sigma(x_2, x) \leq 1$ annehmen. Es gibt nun eine Menge von Vektoren $\{d_1, \dots, d_k\} \in M_D(x_1)$, $k > n$, die eine Basis von \mathcal{V}_2 enthält. Es ist dann

$$x = \sum_{i=1}^k \lambda_i d_i,$$

für alle $\lambda_i > 0$ mit $\sum_{i=1}^k \lambda_i = 1$, wobei (ohne Einschränkung) $d_1 \notin M_D(x_2)$ nach Korollar 3.1.10. Nun gilt

$$\sigma(x_2, x) = \underbrace{\lambda_1 \sigma(x_2, d_1)}_{\geq \lambda_1} + \sum_{i=2}^k \underbrace{\lambda_i \sigma(x_2, d_i)}_{\geq \lambda_i} > \sum_{i=1}^k \lambda_i = 1,$$

was einen Widerspruch zur Annahme $\sigma(x_2, x) \leq 1$ darstellt.

Exaktheit folgt, da die D -VORONOI-Bereiche zu benachbarten D -perfekten Vektoren stets eine gemeinsame Facette haben.

Die Menge $\{x \in P_D \mid d \in M_D(x)\}$ ist für jedes $d \in D \cap \mathcal{V}_2^{\geq 0}$ endlich nach Lemma 3.1.2(vi) und der Tatsache, dass P_D diskret ist. Da die Exaktheit der Kachelung gezeigt ist, folgt somit, dass jeder D -perfekte Vektor x nur endlich viele Nachbarn hat.

Zu zeigen bleibt noch, dass die D -VORONOI-Bereiche $\mathcal{V}_2^{\geq 0}$ vollständig überdecken. Sei dazu $y \in \mathcal{V}_2^{\geq 0}$ und $x \in P_D$. Ist $y \notin \mathcal{D}_D(x)$, so gibt es eine Richtung r von x mit $\sigma(r, y) < 0$. Sei $x_1 := x + \lambda r$ der Nachbar von x in Richtung r . Wegen $\lambda > 0$ ist $\sigma(x_1, y) < \sigma(x, y)$. Dies liefert eine Folge x, x_1, x_2, \dots mit $\sigma(x, y) > \sigma(x_1, y) > \dots$, so dass wieder mit Lemma 3.1.2 und der Diskretheit von P_D folgt, dass sich dieser Prozes nur endlich oft wiederholen lässt. Es gibt daher ein $k \in \mathbb{N}$, so dass $y \in \mathcal{D}_D(x_k)$. \square

Wie in Abschnitt 2.4 erhalten wir hieraus eine wichtige Aussage über den D -VORONOI-Graphen:

Definition 3.1.19. *Der Graph Γ_D mit Knotenmenge P_D und Kantenmenge*

$$E(\Gamma_D) := \{(x, y) \in P_D \times P_D \mid x \text{ und } y \text{ sind Nachbarn}\}$$

heißt der D -VORONOI-Graph.

Korollar 3.1.20. *Sei $D \subset \mathcal{V}_2^{\geq 0} \setminus \{0\}$ diskret und zulässig. Dann ist Γ_D zusammenhängend und lokal endlich, d.h. jeder Knoten in Γ_D hat nur endlich viele Nachbarn.*

BEWEIS. Folgt direkt aus Satz 3.1.18. \square

3.2 Diskontinuierliche Gruppen

In diesem Abschnitt werden diskrete Mengen $D \subset \mathcal{V}_2^{>0} \setminus \{0\}$, die invariant unter Gruppen sind, die diskontinuierlich auf $\mathcal{V}_2^{>0}$ operieren. Wie es in den meisten Computeralgebra-Systemen, so z.B. auch **Magma**, üblich ist, betrachten wir stets Gruppenoperationen von rechts.

Definition 3.2.1. Seien $\mathcal{V}_1^{>0}$ und $\mathcal{V}_2^{>0}$ duale Kegel bezüglich σ .

(i) Die Automorphismengruppe von $\mathcal{V}_i^{>0}$ für $i = 1, 2$ ist definiert durch

$$\text{Aut } \mathcal{V}_i^{>0} := \{g \in \text{GL}(\mathcal{V}_i) \mid \mathcal{V}_i^{>0}g = \mathcal{V}_i^{>0}\}.$$

(ii) Eine Untergruppe $\Omega \leq \text{Aut } \mathcal{V}_i^{>0}$ operiert eigentlich diskontinuierlich auf \mathcal{V}_i , falls folgende zwei Bedingungen erfüllt sind:

1. Der Stabilisator

$$\Omega_x := \text{Stab}_\Omega(x) := \{\omega \in \Omega \mid x\omega = x\}$$

ist für jedes $x \in \mathcal{V}_i^{>0}$ endlich.

2. Für jedes $x \in \mathcal{V}_i^{>0}$ hat die Bahn $x\Omega := \{x\omega \mid \omega \in \Omega\}$ keinen Häufungspunkt in $\mathcal{V}_i^{>0}$.

Man nennt in diesem Fall auch die Gruppe Ω selbst (eigentlich) diskontinuierlich.

Definition 3.2.2. Sei $\Omega \leq \text{Aut } \mathcal{V}_1^{>0}$ und $\omega \in \Omega$. Dann heißt das eindeutig bestimmte Element $\omega^{ad} \in \text{GL}(\mathcal{V}_2)$ mit $\sigma(x\omega, y) = \sigma(x, y\omega^{ad})$ für alle $x \in \mathcal{V}_1$, $y \in \mathcal{V}_2$ das zu ω adjungierte Element. Die Gruppe $\Omega^{ad} := \{\omega^{ad} \mid \omega \in \Omega\} \leq \text{GL}(\mathcal{V}_2)$ heißt die zu Ω adjungierte Gruppe.

Bemerkung 3.2.3. Sind $\mathcal{V}_1^{>0} \subset \mathcal{V}_1$ und $\mathcal{V}_2^{>0} \subset \mathcal{V}_2$ duale Kegel bezüglich σ und ist $g \in \text{GL}(\mathcal{V}_1)$, so sind auch $\mathcal{V}_1^{>0}g$ und $\mathcal{V}_2^{>0}(g^{ad})^{-1}$ duale Kegel bezüglich σ , denn es gilt für $xg \in \mathcal{V}_1$, $y(g^{ad})^{-1} \in \mathcal{V}_2$, dass

$$\sigma(xg, y(g^{ad})^{-1}) = \sigma(x, y(g^{ad})^{-1}g^{ad}) = \sigma(x, y).$$

Um Satz 3.2.7 zu beweisen, wird zusätzlich ist folgende allgemeinere Definition aus der Maßtheorie (vgl. [Vin89, pp. 27-28]) nötig sein.

Definition 3.2.4. Sei V ein endlich-dimensionaler, reeller Vektorraum ausgestattet mit dem LEBESGUE-Maß λ . Der Schwerpunkt einer kompakten Menge $K \subset V$ mit positivem Maß ist definiert durch

$$c(K) := \frac{1}{\lambda(K)} \int_K x \, d\lambda.$$

Lemma 3.2.5. Für ein kompaktes $K \subset V$ mit positivem Maß und jedes $g \in \text{GL}(V)$ gilt $c(gK) = gc(K)$.

BEWEIS. Es gilt

$$\begin{aligned}
c(gK) &= \frac{1}{\lambda(gK)} \int_{gK} x \, d\lambda \\
&= \frac{1}{\det(g)\lambda(K)} \int_K gx \det(g) \, d\lambda \\
&\stackrel{*}{=} \frac{1}{\lambda(K)} g \int_K x \, d\lambda \\
&= gc(K)
\end{aligned}$$

Zu *: Da die Transformation g stetig und linear ist, vertauscht sie mit dem Integral. \square

Angewandt auf die gegebene Situation ergibt sich:

Bemerkung 3.2.6. Für $x \in \mathcal{V}_1^{>0}$ sei $H(x) := \{y \in \mathcal{V}_2^{>0} \mid \sigma(x, y) = 1\}$. Diese Menge ist offenbar konvex und abgeschlossen in $\mathcal{V}_2^{>0}$. Nach Lemma 3.1.2(vi) gilt für $y \in H(x)$, dass $1 = \sigma(x, y) \geq \|y\|_2$, so dass $H(x)$ insgesamt kompakt ist. Sei $x^* := c(H(x))$. Dann definiert $*$ eine stetige Abbildung von $\mathcal{V}_1^{>0}$ nach $\mathcal{V}_2^{>0}$ mit der Eigenschaft

$$(xg)^* = x^* g^{-ad} \text{ für alle } x \in \mathcal{V}_1^{>0}, g \in \text{Aut } \mathcal{V}_1^{>0},$$

was sofort aus Lemma 3.2.5 folgt, denn g^{-ad} ist linear, bijektiv und bildet $H(x)$ nach $H(xg)$ ab.

Satz 3.2.7. Sei $\Omega \leq \text{Aut } \mathcal{V}_1^{>0}$ eine diskontinuierliche Gruppe. Dann operiert auch Ω^{ad} eigentlich diskontinuierlich auf $\mathcal{V}_2^{>0}$.

BEWEIS. Die Behauptung ist offenbar symmetrisch bezüglich Ω und Ω^{ad} . Zur Vereinfachung der Notation sei daher angenommen, dass Ω^{ad} eigentlich diskontinuierlich auf $\mathcal{V}_2^{>0}$ operiert, Ω aber nicht auch $\mathcal{V}_1^{>0}$.

Ist für ein $x \in \mathcal{V}_1^{>0}$ der Stabilisator Ω_x unendlich, so ist auch $(\Omega_x)^{ad}$ unendlich. Wegen $H(x) = H(x\omega)$ für $\omega \in \Omega_x$ und $\sigma(x\omega, y) = \sigma(x, y\omega^{ad})$ ist aber $(\Omega_x)^{ad} \leq \Omega_{x^*}^{ad}$, was nicht sein kann, da Ω^{ad} diskontinuierlich operiert.

Gibt es ein $x \in \mathcal{V}_1^{>0}$, so dass die Bahn $x\Omega$ einen Häufungspunkt $x_0 \in \mathcal{V}_1^{>0}$ besitzt, so ist $x_0^* \in \mathcal{V}_2^{>0}$ ein Häufungspunkt von $x^*\Omega^{ad}$, da $*$ stetig ist. Dies ist ebenfalls ein Widerspruch, also folgt insgesamt die Behauptung. \square

Von nun an sei $\Omega \leq \text{Aut } \mathcal{V}_1^{>0}$ eine eigentlich diskontinuierliche Gruppe und $D \subset \mathcal{V}_2^{>0} \setminus \{0\}$ eine in $\mathcal{V}_2^{>0}$ diskrete Menge, die zudem zulässig und invariant unter der Operation von Ω^{ad} sei.

Lemma 3.2.8. Für jedes $x \in \mathcal{V}_1^{>0}$ und $\omega \in \Omega$ gilt

- (i) $\mu_D(x\omega) = \mu_D(x)$,
- (ii) $M_D(x\omega) = M_D(x)(\omega^{ad})^{-1}$,
- (iii) $\mathcal{D}_D(x\omega) = \mathcal{D}_D(x)(\omega^{ad})^{-1}$.

Insbesondere operiert Ω auf dem Graphen Γ_D .

BEWEIS. Sei $d \in M_D(x)$. Dann gilt

$$\begin{aligned}\mu_D(x) &= \sigma(x, d) = \sigma(x\omega\omega^{-1}, d) \\ &= \sigma(x\omega, d(\omega^{ad})^{-1}) \geq \mu_D(x\omega).\end{aligned}$$

Wendet man nun ω^{-1} auf $x\omega$ an, so erhält man $\mu_D(x\omega) \geq \mu_D(x)$. Daraus folgt (i).

Setzt man in obige Gleichung $d'(\omega^{ad})^{-1}$ mit $d' \in M_D(x\omega)$ anstelle von $d \in M_D(x)$ ein, so ergibt sich (ii) und (iii) folgt direkt aus (ii). \square

Der folgende Satz von BASS und SERRE wird nun zusammen mit den bisherigen Erkenntnissen das wesentliche Hilfsmittel zum Beweis des Hauptresultates in Satz 3.2.10 bilden. An dieser Stelle sei für die nötigen Begriffe aus der Graphentheorie nochmals auf Anhang A verwiesen.

Satz 3.2.9. (BASS, SERRE, 1977)

Sei G eine Gruppe, die auf einem zusammenhängenden Graphen Γ operiere. Weiter sei S ein zusammenhängendes Vertretersystem der G -Nebenklassen von Γ , $(q_e \mid e \in E(S))$ eine verbindende Familie für S und T ein maximaler Teilbaum von S .

Dann gibt es einen zusammenhängenden Gruppengraphen $\mathcal{G} : \Gamma/G \rightarrow \text{Grp}$ definiert durch $\mathcal{G}(\bar{s}) = \text{Stab}_G(s)$ für alle $s \in S$, wobei für $e \in E(S)$ die Abbildungen $\mathcal{G}_e \rightarrow G_{ie}, G_{q_e^{-1}\tau_e}$ jeweils gegeben sind durch $g \mapsto g, g^{q_e}$. Es gibt einen surjektiven Homomorphismus $\phi : \pi := \pi(\mathcal{G}, \bar{T}) \rightarrow G$, $p \mapsto \hat{p}$, der durch die Einbettungsabbildung $\mathcal{G}(\bar{v}) \hookrightarrow G$ und zusammen mit $q_{\bar{e}} \mapsto q_e$ für alle $v \in V(S), e \in E(S)$ eindeutig bestimmt ist.

BEWEIS. [Dic80, Theorem 6.1] \square

Der folgende Satz ist eine Folgerung aus Satz 3.2.9 und liefert die Möglichkeit ein Erzeugendensystem für die Gruppe Ω zu konstruieren. Er liefert die Grundlage für die hier vorgestellten Algorithmen.

Satz 3.2.10. Es sei $D \subset \mathcal{V}_2^{\geq 0} \setminus \{0\}$ zulässig und diskret in \mathcal{V}_2 und $\Omega \leq \text{Aut } \mathcal{V}_1^{\geq 0}$ eine Gruppe, die eigentlich diskontinuierlich auf $\mathcal{V}_1^{\geq 0}$ operiert, so dass D invariant unter Ω^{ad} ist und der Restklassengraph Γ_D/Ω endlich ist. Weiter sei $X := \{x_1, \dots, x_\ell\}$ ein Vertretersystem von D -perfekten Punkten, die einen Teilbaum T von Γ_D erzeugen und $T_1 = \delta(T)$. Für jedes $y \in T_1$ wähle ein $\omega_y \in \Omega$, so dass $\omega_y^{-1}(y) \in X$. Dann gilt

$$\Omega = \langle \omega_y, \text{Stab}_\Omega x_i \mid y \in T_1, i = 1 \dots \ell \rangle.$$

Insbesondere ist Ω endlich erzeugt.

BEWEIS. Als zu betrachtendes Vertretersystem des Restklassengraphen wählt man den von X erzeugten Baum. Nach Satz 3.2.9 ist in diesem Fall Ω epimorphes Bild der Fundamentalgruppe π . Als Erzeuger der Fundamentalgruppe erhält man einfach die Knotengruppen, sowie die Elemente der verbindenden Familie (vgl. A), deren Identifikation mit $\text{Stab}_\Omega(x_i)$ bzw. ω_y aus Satz 3.2.9 folgt. \square

Aus Satz 3.2.10 erhält man unmittelbar folgenden Algorithmus zur Berechnung eines Erzeugendensystems für Ω .

Algorithmus 3.2.11. (OPGENORTH, 2001)

1. Bestimme einen D -perfekten Vektor x_1 . Setze $L_1 := \{x_1\}$, $L_2 := \emptyset$, $S = \emptyset$.
2. Sobald $L_1 = \emptyset$: STOP; sonst wähle $x \in L_1$.
3. Bestimme ein Erzeugendensystem S_x von $\text{Stab}_\Omega(x)$ und setze $S = S \cup S_x$.
4. Berechne die Menge R aller Richtungen von x und die Menge R' ein Vertretersystem der Bahnen von $\text{Stab}_\Omega(x)$ auf R .
5. Bestimme $\delta(R') := \{y \in P_D \mid y \text{ Nachbar von } x \text{ in Richtung } r \in R'\}$.
6. Für $y \in \delta(R')$ prüfe, ob ein $z \in L_1 \cup L_2$ und ein $\omega \in \Omega$ mit $y\omega = z$ existiert.
Falls nein, setze $L_1 = L_1 \cup \{y\}$.
Falls $z \in L_1$, setze $S = S \cup \{\omega\}$.
7. Setze $L_2 = L_2 \cup \{x\}$, $L_1 = L_1 \setminus \{x\}$ und gehe zu Schritt 2.

Dann ist L_2 ein Vertretersystem für P_D/Ω und S erzeugt Ω .

Bemerkung 3.2.12. Satz 3.2.10 lässt sich in vielerlei Situationen anwenden, in denen dann $\mathcal{V}_1^{>0}$ und die diskontinuierliche Gruppe Ω gegeben sind. Konkret sind dabei folgende Aufgaben zu erledigen:

- (i) Man muss einen dualen Kegel $\mathcal{V}_2^{>0}$ und eine geeignete Abbildung σ finden, so dass $\mathcal{V}_1^{>0}$ und $\mathcal{V}_2^{>0}$ duale Kegel bezüglich σ sind.
- (ii) Man finde eine diskrete, zulässige Teilmenge $D \subset \mathcal{V}_2^{>0} \setminus \{0\}$, die unter der Operation von Ω^{ad} invariant ist.
- (iii) Man finde eine Methode, zu $x \in \mathcal{V}_1$ die minimalen Vektoren $M_D(x)$ zu bestimmen.
- (iv) Man berechne die Richtungen von x .
- (v) Man zeige, dass Ω auf P_D nur endlich viele Bahnen hat.
- (vi) Man finde eine Methode, ein Erzeugendensystem für die Stabilisatoren und die ω_y aus Satz 3.2.10 zu bestimmen.

Bemerkung 3.2.13. Selbst wenn man zeigen kann, dass Algorithmus 3.2.11 in einer gegebenen Anwendung stets terminiert, so kann man dennoch seine Laufzeit nicht durch die Größe des Inputs beschränken. Die Laufzeit hängt maßgeblich von der Anzahl der Ω -Restklassen D -perfekter Punkte sowie der Anzahl ihrer Richtungen ab, die beide a priori nicht abzuschätzen sind. So gibt es z.B. für die maximal endlichen Matrixgruppen vom Grad 24 bis auf wenige Ausnahmen nur eine einzige G -perfekte Form und der Normalisator ist endlich. Umgekehrt gibt es vermeintlich kleine Beispiele, für die der Automorphismen-Algorithmus aus Kapitel 5 über 100 inäquivalente perfekte Punkte findet (vgl. Beispiel 6.1.1(3)).

Bemerkung 3.2.14. *Der Stabilisator eines D -perfekten Vektors x permutiert die Richtungen von x . Da diese eine Basis von \mathcal{V}_1 enthalten, ist $\omega \in \text{Stab}_\Omega(x)$ durch diese Permutation eindeutig bestimmt. Allerdings ist diese Methode praktisch ineffizient.*

Für einen wichtigen Spezialfall löst die folgende Proposition das Problem v in Bemerkung 3.2.12.

Proposition 3.2.15. *Sei L ein volles Gitter in \mathcal{V}_2 und $D = L \cap \mathcal{V}_2^{\geq 0} \setminus \{0\}$, so dass $\Omega \leq \text{Aut}(\mathcal{V}_1)$ eigentlich diskontinuierlich auf $\mathcal{V}_1^{\geq 0}$ operiert und D zulässig ist. Falls es eine endliche Teilmenge $T \subset D$ gibt, so dass die Pyramide*

$$C = \left\{ \sum_{v \in T} \alpha_v v \mid \alpha_v \in \mathbb{R}, \alpha_v \geq 0 \right\}$$

einen Fundamentalbereich (vgl. Definition 3.3.1) der Operation von Ω^{ad} auf $\mathcal{V}_2^{\geq 0}$ enthält, dann ist der Restklassengraph Γ_D/Ω endlich.

BEWEIS. Sei $n := \dim \mathcal{V}_2$ und

$$M := \{y \in D \mid y_T - y \in \mathcal{V}_2^{\geq 0}\} \subset D$$

mit $y_T := n \sum_{v \in T} v$. Für $x \in \mathcal{V}_1^{\geq 0}$ und $y \in M$ gilt

$$\sigma(x, y) \leq \sigma(x, y) + \sigma(x, y_T - y) = \sigma(x, y_T),$$

so dass mit Lemma 3.1.2 folgt, dass M endlich ist.

Sei nun $x \in P_D$ und $d_1, \dots, d_n \in M_D(x)$ linear unabhängig. Da der Kegel C einen Fundamentalbereich der Operation von Ω^{ad} auf $\mathcal{V}_2^{\geq 0}$ enthält, kann man ohne Beschränkung der Allgemeinheit annehmen, dass $d_0 = d_1 + \dots + d_n \in C$, also ist $d_0 = \sum_{v \in T} \alpha_v v$ für gewisse $\alpha_v \geq 0$. Damit folgt

$$n = \sigma(x, d_0) = \sum_{v \in T} \alpha_v \sigma(x, v),$$

also ist wegen $\sigma(x, v) \geq 1$ für alle $v \in T$ auch $0 \leq \alpha_v \leq n$, also

$$y_T - d_0 = \sum_{v \in T} (n - \alpha_v) v \in \mathcal{V}_2^{\geq 0}.$$

Zudem hat man für $i = 1, \dots, n$

$$y_T - d_i = y_T - d_0 + \sum_{j \neq i} d_j \in \mathcal{V}_2^{\geq 0},$$

also $d_i \in M$. Insgesamt folgt also, dass für jedes $x \in P_D$ ein $\omega \in \Omega$ existiert, so dass $M \cap M_D(x\omega)$ eine Basis von \mathcal{V}_2 enthält.

Jedes $x \in P_D$ ist jedoch eindeutig durch eine Teilmenge von $M_D(x)$ bestimmt, die eine Basis von \mathcal{V}_2 enthält. Da M aber endlich ist, gibt es nur endlich viele Teilmengen von M , die n linear unabhängige Vektoren enthalten. Damit kann es auch nur endlich viele $x \in P_D$ geben, so dass $M \cap M_D(x)$ n linear unabhängige Vektoren enthält. Damit folgt die Behauptung. \square

3.3 Fundamentalbereiche und Endlichkeitsaussagen

Um die Endlichkeit des Restklassengraphen zu zeigen ist das Studium von Fundamentalbereichen hilfreich. Hierzu werden die Resultate von KOECHER über Fundamentalbereiche (vgl. [Koe60, §5] und [Koe61, §1]) auf duale Kegel verallgemeinert.

Definition 3.3.1. Sei \mathfrak{X} ein lokal kompakter¹ topologischer Raum und $\text{Aut}(\mathfrak{X})$ die Gruppe der Homöomorphismen von \mathfrak{X} nach \mathfrak{X} .

- (i) Eine Untergruppe $\Omega \leq \text{Aut}(\mathfrak{X})$ heißt diskontinuierlich, wenn für jedes $x \in \mathfrak{X}$ die Bahn $x\Omega$ keinen Häufungspunkt in \mathfrak{X} besitzt.
- (ii) Sei Ω eine diskontinuierliche Gruppe und $\mathbb{F} \subset \mathfrak{X}$ eine Teilmenge mit folgenden Eigenschaften:
 - (a) \mathbb{F} ist abgeschlossen,
 - (b) Zu jedem $x \in \mathfrak{X}$ existiert ein $\omega \in \Omega$ mit $x\omega \in \mathbb{F}$.
 - (c) Gilt für $x \in \mathfrak{X}$ und $\psi, \omega \in \Omega$, dass $x\psi, x\omega \in \mathbb{F}^\circ$, so ist $\psi = \omega$.
 - (d) Zu jeder kompakten Menge $K \subset \mathfrak{X}$ gibt es nur endlich viele $\omega \in \Omega$ mit $\mathbb{F}\omega \cap K \neq \emptyset$.

Dann nennt man \mathbb{F} einen Fundamentalbereich von Ω (in \mathfrak{X}).

Sei ab hier wieder spezieller Ω eine auf den bezüglich σ dualen Kegeln $\mathcal{V}_1^{>0}$ und $\mathcal{V}_2^{>0}$ diskontinuierliche Gruppe und $D \subset \mathcal{V}_2^{>0} \setminus \{0\}$ eine diskrete, zulässige Menge, die unter Ω^{ad} invariant ist.

Daraus erhält man bereits ein einfaches Kriterium für die Endlichkeit des Restklassengraphen:

Lemma 3.3.2. Sei $D_0 \subset \mathcal{V}_2^{>0} \setminus \{0\}$ mit der Eigenschaft, dass es ein $y_0 \in \mathcal{V}_2^{>0}$ gibt mit $d - y_0 \in \mathcal{V}_2^{>0}$ für alle $d \in D_0$, was z.B. für endliches D_0 stets erfüllt ist. Ist $D = D_0\Omega$, so ist der Restklassengraph Γ_D/Ω endlich.

BEWEIS. Sei $x \in P_D$ gegeben. Dann existiert nach Voraussetzung ein $\omega \in \Omega$, so dass $M_D(x\omega) = M_D(x)\omega^{-ad}$ (vgl. Lemma 3.2.8) einen Punkt mit D_0 gemeinsam hat. Es reicht daher zu zeigen, dass es nur endlich viele $x \in P_D$ gibt mit $M_D(x) \cap D \neq \emptyset$. Für ein solches x existiert nun ein $d \in D_0$ mit

$$1 = \mu_D(x) = \sigma(x, d) \geq \sigma(x, y_0) \geq \rho(y_0)\|x\|_1,$$

mit $\|\cdot\|_1$ und $\rho(y_0)$ wie in Lemma 3.1.2. Nach Lemma 3.1.9 ist aber P_D eine diskrete Menge, so dass es nur endlich viele $x \in P_D$ mit dieser Eigenschaft geben kann. \square

Die folgende Konstruktion eines Fundamentalbereiches für die Operation von Ω^{ad} auf $\mathcal{V}_2^{>0}$ findet sich für den Fall von selbstdualen Kegeln bereits in [Koe60, Satz 3]. Dazu ist vorher noch ein Lemma nötig.

¹d.h. jede offene Umgebung jedes Punktes $x \in \mathfrak{X}$ enthält eine kompakte Umgebung von x

Lemma 3.3.3. *Sei $K \subset \mathcal{V}_2^{>0}$ kompakt. Dann existieren nur endlich viele $x \in P_D$, so dass $\mathcal{D}_D(x) \cap K \neq \emptyset$.*

BEWEIS. Sei $x \in P_D$ und $y \in K \cap \mathcal{D}_D(x)$. Da x D -perfekt ist, gibt es n linear unabhängige $v_1, \dots, v_n \in M_D(x)$ mit

$$y = \sum_{k=1}^n \lambda_k v_k, \quad \lambda_k \geq 0.$$

Die Norm $\|y\|_2$ ist durch eine von K abhängige Konstante $c_1 > 0$ beschränkt. Da D diskret ist, gibt es außerdem eine Konstante $c_2 > 0$ mit $\|d\|_2 \geq c_2$ für alle $d \in D$. Da für jedes $k \in \{1, \dots, n\}$ gilt, dass $y - \lambda_k v_k \in \mathcal{V}_2^{>0}$, so dass eine weitere Konstante $c_3 > 0$ mit $\|y\|_2 \geq c_3 \lambda_k \|v_k\|_2$. Es folgt

$$c_1 \geq \|y\|_2 \geq c_3 \lambda_k.$$

Nach Lemma 3.1.2 (mit vertauschten Rollen von $\mathcal{V}_1^{>0}$ und $\mathcal{V}_2^{>0}$) hat man also

$$\rho(K) \|x\|_1 \leq \sigma(x, y) = \sum_{k=1}^n \lambda_k \mu_D(x) \leq c_5$$

für eine positive Konstante c_5 , so dass die $x \in P_D$ mit $K \cap \mathcal{D}_D(x) \neq \emptyset$ alle ebenfalls in einer kompakten Menge in $\mathcal{V}_1^{>0}$ liegen. Da nach Lemma 3.1.9 die Menge P_D diskret ist, gibt es also nur endlich viele solche x und das war behauptet. \square

Satz 3.3.4. (KOECHER, 1960)

Für $x \in P_D$ sei $\mathcal{D}_D(x)/\Omega_x^{ad}$ ein Fundamentalbereich der Operation des adjungierten Stabilisators Ω_x^{ad} von x auf dem D -VORONOI-Bereich von x . Bezeichne weiter P_D/Ω eine Transversale der Ω -Äquivalenzklassen D -perfekter Punkte. Definiere

$$\mathbb{F}(D, \Omega) = \bigcup_{x \in P_D/\Omega} \mathcal{D}_D(x)/\Omega_x^{ad}.$$

Falls P_D/Ω endlich ist, so ist die Menge

$$\mathbb{F} := \mathbb{F}(D, \Omega) \cap \mathcal{V}_2^{>0}$$

ein Fundamentalbereich von Ω^{ad} in $\mathcal{V}_2^{>0}$.

BEWEIS. Zunächst existiert $\mathcal{D}_D(x)/\Omega_x^{ad}$ immer, da Ω diskontinuierlich ist und daher der Stabilisator von $x \in P_D$ stets endlich ist. Außerdem operiert Ω_x^{ad} auf $\mathcal{D}_D(x)$.

Als Vereinigung von endlich vielen abgeschlossenen Mengen ist \mathbb{F} abgeschlossen.

Da die D -VORONOI-Bereiche der D -perfekten Punkte den Kegel $\mathcal{V}_2^{>0}$ überdecken, gibt es zu jedem $y \in \mathcal{V}_2^{>0}$ ein $x \in P_D$ mit $y \in \mathcal{D}_D(x)$. Dann gibt es ein $\omega \in \Omega$ mit $\tilde{x} = x\omega \in P_D/\Omega$ und es gilt dann $y\omega^{-ad} \in \mathcal{D}_D(x)\omega^{-ad} = \mathcal{D}_D(\tilde{x})$. Dann gibt es ein $\psi \in \Omega_{\tilde{x}}^{ad}$ mit $y\omega^{-ad}\psi \in \mathcal{D}_D(\tilde{x})/\Omega_{\tilde{x}}^{ad} \cap \mathcal{V}_2^{>0} \subset \mathbb{F}$.

Angenommen, es gibt eine offene Menge $\emptyset \neq U \subset \mathcal{V}_2^{>0}$, so dass \bar{U} in $\mathcal{V}_2^{>0}$ kompakt ist und $U \subset \mathbb{F}$ und $U\omega^{ad} \subset \mathbb{F}$ für ein $\omega \in \Omega$. Nach Lemma 3.3.3 gibt es dann endlich viele D -perfekte Punkte $x_1, \dots, x_k, x'_1, \dots, x'_\ell \in P_D$ mit

$$U \subset \bigcup_{i=1}^k \mathcal{D}_D(x_i) \quad \text{und} \quad U\omega^{ad} \subset \bigcup_{j=1}^{\ell} \mathcal{D}_D(x'_j).$$

Daraus folgt dann

$$\begin{aligned} U &\subseteq \left(\bigcup_{i=1}^k \mathcal{D}_D(x_i) \right) \cap \left(\bigcup_{j=1}^{\ell} \mathcal{D}_D(x'_j\omega^{-1}) \right) \\ &\subseteq \bigcup_{i=1}^k \bigcup_{j=1}^{\ell} (\mathcal{D}_D(x_i) \cap \mathcal{D}_D(x'_j\omega^{-1})). \end{aligned}$$

Da U aber offen war, muss einer der Durchschnitte $\mathcal{D}_D(x_i) \cap \mathcal{D}_D(x'_j\omega^{-1})$ innere Punkte besitzen. Die Kachelung von $\mathcal{V}_2^{>0}$ durch die D -VORONOI-Bereiche ist nach Satz 3.1.18 aber exakt, d.h. zwei D -VORONOI-Bereiche haben genau dann gemeinsame innere Punkte, wenn sie gleich sind. Damit ist dann auch $x_i = x'_j\omega^{-1}$. Da die Punkte aus P_D/Ω paarweise inäquivalent sind, folgt sogar $x_i = x'_j =: x$ und $\omega \in \Omega_x$. Wählt man nun U hinreichend klein, so kann man ohne Einschränkung annehmen, dass $U \subset \mathcal{D}_D(x)/\Omega_x^{ad}$ und $U\omega \subset \mathcal{D}_D(x)/\Omega_x^{ad}$. Nun ist aber $\mathcal{D}_D(x)/\Omega_x^{ad}$ ein Fundamentalbereich, also muss $\omega = 1$ gelten.

Lemma 3.3.3 liefert unmittelbar die Gültigkeit der letzten ausstehenden Bedingung aus Definition 3.3.1. \square

Im Fall, dass D zulässig ist und in einem vollen Gitter in \mathcal{V}_2 enthalten ist, kann man noch einiges mehr aussagen. In [Koe60] gibt KOECHER den folgenden Satz zwar an, beweist jedoch nur eine Richtung der Äquivalenz.

Satz 3.3.5. *Sei L ein volles Gitter in \mathcal{V}_2 und $D \subseteq \mathcal{V}_2^{>0} \cap L \setminus \{0\}$ zulässig. Weiter sei Ω eine diskontinuierliche Gruppe mit der zusätzlichen Eigenschaft, dass $L\Omega^{ad} = L$ gilt und damit $D\Omega^{ad} = D$. Dann sind folgende Aussagen äquivalent:*

- (1) *Es gibt nur endlich viele Äquivalenzklassen D -perfekter Punkte mit D -Minimum 1.*
- (2) *Es gibt eine Menge $\tilde{\mathbb{F}} \subset \mathcal{V}_2^{>0}$, so dass für alle $y \in \mathcal{V}_2^{>0}$ ein $\omega \in \Omega$ existiert mit $y\omega^{ad} \in \tilde{\mathbb{F}}$, mit der Eigenschaft, dass es ein $y_0 \in \mathcal{V}_2^{>0}$ gibt mit $d - y_0 \in \mathcal{V}_2^{>0}$ für alle $d \in \tilde{\mathbb{F}} \cap L$.*

BEWEIS. (1) \Rightarrow (2): Definiere $\tilde{\mathbb{F}} = \mathbb{F}$ aus Satz 3.3.4. Da \mathbb{F} nach Voraussetzung in der Vereinigung endlich vieler D -VORONOI-Bereiche enthalten ist, reicht es, die folgende Behauptung zu zeigen:

Zu jedem $x \in P_D$ existiert ein $y_0 = y_0(x) \in \mathcal{V}_2^{>0}$, so dass $d - y_0 \in \mathcal{V}_2^{>0}$ für alle $d \in \mathcal{D}_D(x) \cap \mathcal{V}_2^{>0} \cap L$.

Dazu bemerkt man zunächst, dass für $x \in P_D$ $\mathcal{D}_D(x)$ eine endliche Vereinigung von Pyramiden der Form

$$P(M) = \left\{ \sum_{v \in M} \lambda_v v \mid \lambda_v \geq 0 \right\},$$

wobei M aus $n = \dim \mathcal{V}_2$ linear unabhängigen Vektoren aus $M_D(x)$ besteht (M existiert, da x D -perfekt ist). Ist $|M_D(x)| = n$, so ist diese Behauptung klar, ansonsten gibt es eine Hyperebene durch 0 , die $\mathcal{D}_D(x)$ in zwei Pyramiden $P(M)$ und $P(M')$ aufteilt, M wie oben. M' hat dann auf jeden Fall wenigstens ein Element weniger als $M_D(x)$, so dass die Behauptung durch Induktion folgt.

Die Behauptung reduziert sich so nochmals darauf, für solche Pyramiden $P(M)$ mit M wie oben ein $y_0(M) \in \mathcal{V}_2^{>0}$ zu finden mit $d - y_0(M) \in \mathcal{V}_2^{\geq 0}$ für alle $d \in P(M) \cap \mathcal{V}_2^{>0} \cap L$. Denn dann gibt es auch ein y_0 mit $y_0(M) - y_0 \in \mathcal{V}_2^{\geq 0}$ für alle vorkommenden M . Sei also $M = \{v_1, \dots, v_n\}$ und $L_M := \langle M \rangle_{\mathbb{Z}}$ das von M erzeugte (volle) Teilgitter von L . Der Exponent γ der endlichen abelschen Gruppe L/L_M erfüllt dann $\gamma L \subseteq L_M$, also gilt für alle $\ell \in L \cap P(M) \cap \mathcal{V}_1^{>0}$:

$$\gamma \ell = \sum_{k=1}^n \lambda_k v_k,$$

mit $\lambda_k \in \mathbb{N}_0$ und nicht sämtlich 0 . Sei

$$y_\ell := \frac{1}{\gamma} \sum_{\substack{k=1 \\ \lambda_k \neq 0}}^n v_k.$$

Dann ist offenbar $\ell - y_\ell \in \mathcal{V}_2^{\geq 0}$ und $y_\ell \in \mathcal{V}_2^{\geq 0}$. Wäre nun $y_\ell \in \partial \mathcal{V}_2^{>0}$, so gäbe es ein $x' \in \mathcal{V}_1^{\geq 0} \setminus \{0\}$ mit $\sigma(x', y_\ell) = 0$. Dann wäre aber auch $\sigma(x', v_k) = 0$ und damit $\sigma(x', \ell) = 0$. Das kann aber nach der Wahl von ℓ nicht sein.

Es gibt zu M offenbar höchstens 2^n Vektoren der Form y_ℓ , es gibt also ein $y_0(M) \in \mathcal{V}_2^{\geq 0}$ mit $y_\ell - y_0(M) \in \mathcal{V}_2^{\geq 0}$ für alle ℓ . Damit folgt die Behauptung.

(2) \Rightarrow (1): Definiere $D_0 := \tilde{\mathbb{F}} \cap D \subseteq \tilde{\mathbb{F}} \cap L$. Für jedes $\omega \in \Omega$ und $d \in D$ ist $d\omega^{ad} \in D$, also existiert nach Voraussetzung an $\tilde{\mathbb{F}}$ für jedes $d \in D$ ein $\omega \in \Omega$ mit

$$d\omega^{ad} \in D_0.$$

Nach Voraussetzung existiert aber ein $y_0 \in \mathcal{V}_2^{>0}$ mit $d - y_0 \in \mathcal{V}_2^{\geq 0}$ für alle $d \in D_0$, sodass die Behauptung mit Lemma 3.3.2 folgt. \square

Bemerkung 3.3.6. Satz 3.3.5 liefert einen weiteren, etwas allgemeineren Ansatz als Proposition 3.2.15, um in dem Spezialfall, dass die betrachtete zulässige Menge D in einem Gitter enthalten ist, zu zeigen, dass der Restklassengraph Γ_D/Ω endlich ist, Algorithmus 3.2.11 also terminiert:

Ist man in der Lage, a priori hinreichend viele Elemente von Ω zu bestimmen, ohne den Algorithmus zu bemühen, kann man ggf. eine Obermenge des Fundamentalbereiches bestimmen, die die Eigenschaft aus Satz 3.3.5(2) erfüllt.

Kapitel 4

Der Normalisator-Algorithmus

4.1 BRAVAIS-Gruppen

Definition 4.1.1. Sei G eine Untergruppe von $\mathrm{GL}_n(\mathbb{Z})$.

(i) Der Normalisator $N_{\mathbb{Z}}(G)$ von G ist definiert durch

$$N_{\mathbb{Z}}(G) := \{h \in \mathrm{GL}_n(\mathbb{Z}) \mid h^{-1}Gh = G\}.$$

(ii) Es ist

$$\mathcal{F}(G) := \{A \in \mathbb{R}_{sym}^{n \times n} \mid gAg^{tr} = A \text{ f\"ur alle } g \in G\}$$

der Raum der G -invarianten Formen. Die Menge der positiv semidefiniten Elemente von $\mathcal{F}(G)$ wird mit $\mathcal{F}^{\geq 0}(G)$ bezeichnet, die Menge der positiv definiten Elemente mit $\mathcal{F}^{> 0}(G)$. Man nennt $\mathcal{F}^{> 0}(G)$ auch die BRAVAIS-Mannigfaltigkeit von G .

(iii) Sei $X \subseteq \mathbb{R}_{sym}^{n \times n}$. Dann ist die BRAVAIS-Gruppe von X definiert durch

$$B(X) := \{g \in \mathrm{GL}_n(\mathbb{Z}) \mid gAg^{tr} = A \text{ f\"ur alle } A \in X\}.$$

(iv) Die BRAVAIS-Gruppe von G ist definiert als $\mathcal{B}(G) := B(\mathcal{F}(G))$. Gilt $G = \mathcal{B}(G)$, so nennt man G selbst eine BRAVAIS-Gruppe.

Bemerkung 4.1.2. Die BRAVAIS-Mannigfaltigkeit $\mathcal{F}^{> 0}(G)$ von G ist genau dann nicht leer, wenn G eine endliche Gruppe ist.

Ist n\u00e4mlich G endlich und $A \in \mathbb{R}_{sym}^{n \times n}$ eine beliebige, positiv definite Matrix, so ist f\u00fcr jedes $g \in G$ auch gAg^{tr} positiv definit und damit auch

$$inv(A) := \frac{1}{|G|} \sum_{g \in G} gAg^{tr}.$$

Diese Matrix ist aber offenbar auch G -invariant, also gilt $inv(A) \in \mathcal{F}^{> 0}(G)$.

Die umgekehrte Implikation folgt direkt aus Lemma 4.1.3.

Lemma 4.1.3. *Die volle unimodulare Gruppe $GL_n(\mathbb{Z})$ (und damit auch jede unimodulare Gruppe) operiert eigentlich diskontinuierlich auf der Menge der positiv definiten quadratischen Formen*

$$\mathcal{Q}^{>0} := \{Q \in \mathcal{Q} \mid Q \text{ positiv definit}\}$$

bzw. nach Wahl einer Basis auf $\mathbb{R}_{>0}^{n \times n}$ vermöge

$$\mathbb{R}_{>0}^{n \times n} \times G \rightarrow \mathbb{R}_{>0}^{n \times n}, (Q, g) \mapsto gQg^{tr}.$$

BEWEIS. Jede positiv definite Form Q definiert ein volles Gitter L im EUKLIDISCHEN Raum $(\mathbb{R}^n, (\cdot, \cdot))$. Die Operation von $GL_n(\mathbb{Z})$ entspricht einem Wechsel der Gitterbasis, also entspricht die Bahn $Q GL_n(\mathbb{Z})$ von Q der Menge aller Gitterbasen von L . Da L aber eine in \mathbb{R}^n diskrete Menge ist, kann diese Bahn offenbar keinen Häufungspunkt haben.

Um zu zeigen, dass der Stabilisator einer gegebenen Form unter $GL_n(\mathbb{Z})$ stets endlich ist, fasst man diesen als die Automorphismengruppe des durch Q definierten Gitters auf. Diese ist nach Satz 2.1.10 endlich, also folgt die Behauptung. \square

Der Beweis des folgenden Lemmas basiert auf [BNZ73].

Lemma 4.1.4. *Sei $G \leq GL_n(\mathbb{Z})$ endlich.*

- (i) *Die BRAVAIS-Gruppe $\mathcal{B}(G)$ von G ist endlich und es gilt $\mathcal{B}(\mathcal{B}(G)) = \mathcal{B}(G)$, so dass $\mathcal{B}(G)$ selbst eine BRAVAIS-Gruppe ist.*
- (ii) *$\mathcal{F}^{>0}(G)$ ist ein offener Kegel in $\mathcal{F}(G)$.*
- (iii) *$N_{\mathbb{Z}}(G)$ operiert eigentlich diskontinuierlich auf $\mathcal{F}^{>0}(G)$ vermöge $A \mapsto h^{-1}Ah^{-tr}$ für $h \in N_{\mathbb{Z}}(G)$.*
- (iv) *Es gilt $N_{\mathbb{Z}}(\mathcal{B}(G)) = \{h \in GL_n(\mathbb{Z}) \mid h^{-1}\mathcal{F}(G)h = \mathcal{F}(G)\}$.*
- (v) *Der Index $[N_{\mathbb{Z}}(\mathcal{B}(G)) : N_{\mathbb{Z}}(G)]$ ist endlich.*

BEWEIS. Offensichtlich gilt allgemein für $G \leq H$, dass $\mathcal{F}(H) \leq \mathcal{F}(G)$ als \mathbb{R} -Vektorräume. Nach Definition ist $G \leq \mathcal{B}(G)$ und $\mathcal{F}(\mathcal{B}(G)) = \mathcal{F}(G)$. Damit ist aber $\mathcal{B}(\mathcal{B}(G)) = \mathcal{B}(G)$ und mit Bemerkung 4.1.2 ist $\mathcal{B}(G)$ genau dann endlich, wenn G endlich ist, also folgt (i). Für (ii) und (iii) ist nichts zu zeigen und (iv) zeigt sich durch folgende Rechnung:

Sei $h \in N_{\mathbb{Z}}(\mathcal{B}(G))$, $b \in \mathcal{B}(G)$ und $A \in \mathcal{F}(G)$ beliebig. Dann ist

$$A = (h^{-1}bh)A(h^{-1}bh)^{tr} = (h^{-1}b) \underbrace{(hAh^{tr})}_{\in \mathcal{F}(G)} (h^{-1}b)^{tr},$$

also $hAh^{tr} = b(hAh^{tr})b^{tr}$, also $h\mathcal{F}(G)h^{tr} = \mathcal{F}(G)$. Da $\mathcal{B}(G)$ eine BRAVAIS-Gruppe ist, gilt auch die Umkehrung, also (iv).

Damit folgt auch, dass $N_{\mathbb{Z}}(G) \leq N_{\mathbb{Z}}(\mathcal{B}(G))$. Da $\mathcal{B}(G)$ endlich ist, enthält sie nur endlich viele Untergruppen, die \mathbb{Z} -äquivalent zu G sind. Demnach ist die Bahn $G^{N_{\mathbb{Z}}(\mathcal{B}(G))}$ endlich. Dies liefert eine Permutationsdarstellung von $N_{\mathbb{Z}}(\mathcal{B}(G))$ und $N_{\mathbb{Z}}(G)$ ist genau der Stabilisator von G , so dass der Index $[N_{\mathbb{Z}}(\mathcal{B}(G)) : N_{\mathbb{Z}}(G)]$ endlich sein muss. \square

4.2 Der Algorithmus

Um nun den eher allgemeinen Algorithmus 3.2.11 konkret zur Berechnung von Normalisatoren endlicher unimodularer Gruppen zu nutzen, werden nun sukzessiv die Punkte aus Bemerkung 3.2.12 abgearbeitet. Zunächst zur Konstruktion der dualen Kegel und einer geeigneten Abbildung σ :

Lemma 4.2.1. *Sei $G \leq \mathrm{GL}_n(\mathbb{Z})$ eine endliche unimodulare Gruppe. Dann sind $\mathcal{F}^{>0}(G)$ und $\mathcal{F}^{>0}(G^{tr})$ duale Kegel bezüglich*

$$\sigma : \mathcal{F}(G) \times \mathcal{F}(G^{tr}) \rightarrow \mathbb{R}, (A, B) \mapsto \mathrm{Spur}(A \cdot B).$$

BEWEIS. Das Axiom (DC1) wurde bereits in Lemma 4.1.4(ii) nachgeprüft.

Zu Axiom (DC2) sei bemerkt, dass $\mathbb{R}_{>0}^{n \times n}$ einen selbstdualen Kegel bezüglich σ bildet (vgl. [Koe60]). Insbesondere ist daher $\mathrm{Spur}(A \cdot B) > 0$ für alle $A \in \mathcal{F}^{>0}(G)$, $B \in \mathcal{F}^{>0}(G^{tr})$, sodass (DC2) folgt.

Zum Beweis von (DC3) bedient man sich derselben Taktik wie in Bemerkung 4.1.2: Da $\mathbb{R}_{>0}^{n \times n}$ ein selbstdualer Kegel bezüglich σ ist, gibt es zu $A \in \mathcal{F}(G) \setminus \mathcal{F}^{>0}(G)$ ein $B \in \mathbb{R}_{>0}^{n \times n}$ mit $\mathrm{Spur}(A \cdot B) < 0$. Setze dann

$$B' := \frac{1}{|G|} \sum_{g \in G} g^{tr} B g.$$

Offenbar ist $B' \in \mathcal{F}^{>0}(G^{tr})$ und es gilt

$$\begin{aligned} \sigma(A, B') &= \frac{1}{|G|} \sum_{g \in G} \mathrm{Spur}(A \cdot g^{tr} B g) = \frac{1}{|G|} \sum_{g \in G} \mathrm{Spur}(g A g^{tr} B) \\ &= \frac{1}{|G|} \sum_{g \in G} \mathrm{Spur}(A \cdot B) = \sigma(A, B) < 0, \end{aligned}$$

was (DC3) impliziert. □

Bemerkung 4.2.2. *Die im Beweis zu Lemma 4.2.1 verwendete Abbildung*

$$\mathrm{inv}^{tr} : \mathbb{R}_{sym}^{n \times n} \rightarrow \mathcal{F}(G^{tr}), A \mapsto \frac{1}{|G|} \sum_{g \in G} g^{tr} A g$$

ist eine Projektion auf $\mathcal{F}(G^{tr})$, die positiv definite Formen auf positiv definite Formen abbildet und zusätzlich

$$\mathrm{Spur}(A \cdot B) = \mathrm{Spur}(A \cdot \mathrm{inv}^{tr}(B))$$

für alle $A \in \mathcal{F}(G)$ und $B \in \mathbb{R}_{sym}^{n \times n}$ erfüllt.

Sei ab jetzt für dieses Kapitel $G \leq \mathrm{GL}_n(\mathbb{Z})$ eine endliche unimodulare Gruppe und

$$D := \{q_x := \mathrm{inv}^{tr}(x^{tr} x) \mid x \in \mathbb{Z}^n\}. \quad (4.1)$$

Bemerkung 4.2.3. *(i) Für das D-Minimum einer Form $A \in \mathcal{F}(G)$ gilt $\mu_D(A) = \mu(A)$, wobei hier mit $\mu(A)$ wie in Kapitel 2 das Minimum von A als quadratische Form gemeint ist.*

(ii) Es gilt $M_D(A) = \text{inv}^{\text{tr}}(M(A))$.

(iii) Die Menge D aus Gleichung (4.1) ist zulässig.

BEWEIS. Es gilt für $A \in \mathcal{F}(G)$ und $x \in \mathbb{Z}^n$

$$\sigma(A, q_x) = \text{Spur}(Ax^{\text{tr}}x) = \text{Spur}(xAx^{\text{tr}}) = xAx^{\text{tr}},$$

also gelten die Behauptungen in (i) und (ii).

Sei $(A_n)_{n \in \mathbb{N}}$ eine Folge in $\mathcal{F}^{>0}(G)$ mit Grenzwert $A \in \partial\mathcal{F}^{>0}(G)$. Damit konvergiert $\det(A_n)$ gegen 0. Nach der HERMITE-Ungleichung in Lemma 2.3.3 gibt es eine Konstante $c > 0$, die nur von n abhängt, so dass

$$\mu(B)^n \leq c \det B$$

für jedes $B \in \mathbb{R}_{>0}^{n \times n}$. Damit muss dann auch $\mu(A_n)$ gegen 0 konvergieren, so dass D in der Tat zulässig ist. \square

Definition 4.2.4. Eine Matrix $A \in \mathcal{F}(G)$ heißt G -perfekt, falls sie D -perfekt ist für D aus Gleichung (4.1) ist, falls also $M_D(A)$ eine Basis von $\mathcal{F}^{>0}(G^{\text{tr}})$ enthält.

Man nennt A G -extrem, falls die Einschränkung der HERMITE-Funktion

$$\gamma : \mathbb{R}_{>0}^{n \times n} \rightarrow \mathbb{R}, A \mapsto \frac{\mu(A)}{\det(A)^n}$$

(vgl. Definition 2.4.1) auf $\mathcal{F}^{>0}(G^{\text{tr}})$ in A ein lokales Maximum annimmt.

Der folgende Satz bildet den Abschluss der theoretischen Betrachtungen zum Algorithmus.

Satz 4.2.5. (i) Bis auf die Operation von $N_{\mathbb{Z}}(G)$ gibt es nur endlich viele G -perfekte Formen, der Restklassengraph ist also endlich.

(ii) Eine Matrix $A \in \mathcal{F}^{>0}(G)$ ist genau dann G -extrem, wenn sie G -perfekt und eutaktisch (siehe Satz 2.4.3) ist.

BEWEIS. zu (i): [JC95, Théorème 5.1]

zu (ii): [BM91] \square

Bemerkung 4.2.6. Setzt man $G = \{I_n\}$ in Satz 4.2.5, so erhält man genau die klassischen Resultate von VORONOI aus Satz 2.4.4 und Satz 2.4.3.

Bemerkung 4.2.7. Sei $G = \mathcal{B}(G)$ eine BRAVAIS-Gruppe und $A \in \mathcal{F}^{>0}(G)$ eine G -perfekte Matrix. Dann enthält die Menge der Richtungen \mathcal{R} von A eine Basis R_1, \dots, R_k des Formenraumes $\mathcal{F}(G^{\text{tr}})$, so dass der Stabilisator von A genau gegeben ist durch

$$\text{Stab}_{N_{\mathbb{Z}}(G)}(A) = \{g \in \text{Aut}(A) \mid gR_i g^{\text{tr}} \in \mathcal{R}, i = 1 \dots k\}.$$

Da für eine beliebige Gruppe G gilt, dass $\mathcal{F}(G) = \mathcal{F}(\mathcal{B}(G))$, kann in diesem Falle die $N_{\mathbb{Z}}(G)$ bestimmt werden, indem man $N_{\mathbb{Z}}(\mathcal{B}(G))$ unter Verwendung obiger Überlegung berechnet und dann, da nach Lemma 4.1.4(v) der Index $[N_{\mathbb{Z}}(\mathcal{B}(G)) : N_{\mathbb{Z}}(G)]$ endlich ist, mittels Bahnenalgorithmus Erzeuger für $N_{\mathbb{Z}}(G)$ berechnet.

Kapitel 5

Automorphismengruppen

Das Hauptziel dieses Kapitels wird es sein, die Methoden aus Kapitel 3 und Kapitel 4 zu verwenden, um Automorphismengruppen hyperbolischer Gitter zu bestimmen.

5.1 Hyperbolische Gitter und duale Kegel

Das erste Ziel dieses Abschnitts wird es sein, geeignete duale Kegel für die Anwendung der Resultate aus Kapitel 3 zu finden. Als Vorbereitung dazu dient das folgende Lemma.

Lemma 5.1.1. *Es sei ab jetzt stets*

$$H_n := \text{diag}(-1, 1, \dots, 1) \in \mathbb{R}^{n \times n}.$$

Dann ist die Menge

$$\mathcal{C} := \{x = (x_1, \dots, x_n) \mid xH_nx^{tr} < 0 \text{ und } x_1 > 0\}$$

ein selbstdualer Kegel bezüglich des Standardskalarprodukts.

BEWEIS. (DC1) Da $e_1 = (1, 0, \dots, 0) \in \mathcal{C}$, ist \mathcal{C} nicht leer.

(DC2) Seien $x = (x_1, \dots, x_n), y = (y_1, \dots, y_n) \in \mathcal{C}$. Zu zeigen ist, dass $xy^{tr} > 0$ gilt. Durch Multiplikation mit einer positiven reellen Zahl kann man zunächst

ohne Einschränkung $x_1 = y_1$ annehmen. Dann gilt:

$$\begin{aligned}
x \cdot y^{tr} &= x_1 y_1 + \sum_{i=2}^n x_i y_i \\
&= x_1 y_1 + \sum_{i=2}^n \frac{1}{2} ((x_i + y_i)^2 - x_i^2 - y_i^2) \\
&\stackrel{*}{=} \frac{1}{2} (x_1^2 + y_1^2) + \sum_{i=2}^n \frac{1}{2} ((x_i + y_i)^2 - x_i^2 - y_i^2) \\
&= \frac{1}{2} (x_1^2 - \sum_{i=2}^n x_i^2) + \frac{1}{2} (y_1 - \sum_{i=2}^n y_i^2 + \sum_{i=2}^n \frac{1}{2} (x_i^2 + y_i^2)) \\
&= \underbrace{-x H_n x^{tr}}_{>0} + \underbrace{(-y H_n y^{tr})}_{>0} + \underbrace{\sum_{i=2}^n \frac{1}{2} (x_i^2 + y_i^2)}_{\geq 0} \\
&> 0.
\end{aligned}$$

Man beachte in $*$, dass nach Annahme $x_1 = y_1$ gilt.

(DC3) Sei $x = (x_1, \dots, x_n) \notin C$. Es ist zu zeigen, dass es dann ein $y \in \bar{C}$ gibt mit $xy^{tr} \leq 0$. Ist $x_1 \leq 0$, dann wähle man $y = (1, 0, \dots, 0)$. Ist $x_1 > 0$, dann ist $-x_1^2 + s \geq 0$, wobei $s := \sum_{i=2}^n x_i^2$. Offenbar ist $s > 0$, so dass analog wie oben (gegebenenfalls nach Multiplikation mit einer positiven reellen Zahl) $s = 1$ angenommen werden kann. Damit ist $0 < x_1 \leq 1$. Da für $y = (1, -x_2, \dots, -x_n)$ gilt, dass $y H_n y^{tr} = -1 + s = 0$, also $y \in \bar{C}$, und $xy^{tr} = x_1 - 1 \leq 0$, folgt die Behauptung. \square

Den allgemeinen Fall führt man mithilfe von Satz 2.1.1 auf den oben diskutierten zurück:

Lemma 5.1.2. Sei $A \in \mathbb{R}_{hyp}^{n \times n}$ und $T \in \text{GL}_n(\mathbb{R})$ mit $TAT^{tr} = H_n$. Definiere $y_1 := e_1 T^{-tr}$, $y_2 := e_1 T$. Dann sind

$$\begin{aligned}
\mathcal{V}_1^{>0} &:= \{x \in \mathbb{R}^{1 \times n} \mid x A x^{tr} < 0 \text{ und } x y_1^{tr} > 0\} \text{ und} \\
\mathcal{V}_2^{>0} &:= \{x \in \mathbb{R}^{1 \times n} \mid x A^{-1} x^{tr} < 0 \text{ und } y_2 x^{tr} > 0\}
\end{aligned}$$

duale Kegel bezüglich des Standardskalarprodukts.

BEWEIS. Offenbar ist $\mathcal{V}_1^{>0} = CT^{-1}$ und $\mathcal{V}_2^{>0} = CT^{tr}$. Nach Bemerkung 3.2.3 sind demnach CT und CT^{-tr} mit C aus Lemma 5.1.1 duale Kegel bezüglich des Standardskalarprodukts. \square

Bemerkung 5.1.3. Man beachte in Lemma 5.1.2, dass diese Definition von der Wahl von T abhängt, denn z.B. ist mit T mit definiertem Kegel $\mathcal{V}_1^{>0}$ auch $-T$ eine geeignete Matrix, die aber sicherlich einen anderen Kegel $\mathcal{V}'^{>0}$ definiert als T . Um genau zu sein gilt in diesem Fall $x \in \mathcal{V}_1^{>0} \Leftrightarrow -x \in \mathcal{V}'^{>0}$.

Da diese Wahl aber nur auf die konkreten dualen Kegel Einfluss nimmt, die Resultate aus Kapitel 3 aber unabhängig von der Wahl der dualen Kegel sind, beeinflusst dies das Endergebnis des Algorithmus nicht.

Es sei daher ab jetzt für eine feste hyperbolische Matrix A eine entsprechende Transformationsmatrix T und damit die Vektoren y_1, y_2 festgelegt.

5.2 Zulässigkeit

Ab jetzt definieren wir für den Rest dieses Kapitels

$$D := \mathbb{Z}^n \cap (\mathcal{V}_2^{\geq 0} \setminus \{0\}). \quad (5.1)$$

Bemerkung 5.2.1. Da für alle $A \in \mathbb{Z}_{hyp}^{n \times n}$ offenbar $-I_n \in \text{Aut}(A)$ gilt, operiert $\text{Aut}_{\mathbb{Z}}(A)$ nicht auf $\mathcal{V}_1^{\geq 0}$ bzw. $\text{Aut}(A)^{tr}$ nicht auf $\mathcal{V}_2^{\geq 0}$.

Allerdings operiert $\Omega := \text{Aut}_{\mathbb{Z}}(A)$ auf $\mathcal{V}_1^{\geq 0} \cup -\mathcal{V}_1^{\geq 0}$. Der Stabilisator N von $\mathcal{V}_1^{\geq 0}$ ist dann ein Normalteiler vom Index 2 in Ω .

Mit den Methoden aus Kapitel 3 lässt sich also ein Erzeugendensystem S für N berechnen, eines für $\text{Aut}(A)$ ist dann gegeben durch $S \cup \{-I_n\}$. Hierbei ist es jedoch im Endeffekt unerheblich für die Berechnung von Ω , ob man die Stabilisatoren perfekter Punkte bzw. die verbindenden Elemente (vgl. Satz 3.2.10) in N ausrechnet und dann das Erzeugendensystem ergänzt oder sie gleich in Ω ausrechnet.

Der Beweis, dass D zulässig ist, benötigt etwas Vorbereitung. Ein Hilfsmittel ist der wohlbekannte MINKOWSKISCHE Gitterpunktsatz.

Satz 5.2.2. (MINKOWSKI, 1905)

Sei L ein volles Gitter in \mathbb{R}^n und $X \subseteq \mathbb{R}^n$ eine konvexe, zentralsymmetrische Menge, d.h. für alle $x, y \in X$ und alle $\lambda \in [0, 1]$ gilt $\lambda x + (1 - \lambda)y \in X$ und es gilt die Implikation $x \in X \Rightarrow -x \in X$. Erfüllt X die Ungleichung

$$\text{vol}(X) \geq 2^n \text{vol}(\mathbb{R}^n/L),$$

so ist $L \cap X \neq \{0\}$.

BEWEIS. [Neu07, Satz 4.4] □

Korollar 5.2.3. Für jedes $\varepsilon > 0$ und jedes $x \in \mathbb{R}^n$ enthält die Menge

$$P_\varepsilon(x) := \{y \in \mathbb{R}^n \mid 0 \leq (x, y) \leq \varepsilon\}$$

einen Punkt $\neq 0$ mit ganzzahligen Koordinaten.

BEWEIS. Seien $x \in \mathbb{R}^n$ und $\varepsilon > 0$ beliebig. Die Menge

$$X = \{y \in \mathbb{R}^n \mid -\varepsilon \leq (x, y) \leq \varepsilon\}$$

ist zentralsymmetrisch und konvex. Außerdem hat X unendliches Volumen, so dass X nach Satz 5.2.2 einen Gitterpunkt $\ell \neq 0$ enthält. Ist $(x, \ell) \geq 0$, so ist $\ell \in P_\varepsilon(x)$, ansonsten betrachte $-\ell$. □

Sehr wesentlich für den Beweis der Zulässigkeit von D aus Gleichung (5.1) wird folgendes Resultat von LOVÁSZ sein. Dazu zunächst folgende Definitionen.

Definition 5.2.4. Sei $S \subset \mathbb{R}^n$ konvex. Man nennt dann S eine maximal konvexe, gitterpunktfreie Menge, kurz MKGF-Menge, falls gilt:

- (1) Das relative Innere von S (bezüglich $\langle S \rangle_{\mathbb{R}}$) enthält keine ganzzahligen Punkte
- (2) Für $T \subset \mathbb{R}^n$ konvex mit $T \cap \mathbb{Z}^n = \emptyset$ gilt die Implikation

$$S \subseteq T \Rightarrow S = T,$$

das heißt S ist maximal bezüglich Inklusion.

Definition 5.2.5. Sei $V \subseteq \mathbb{R}^n$ ein affiner Teilraum. Man nennt V rational, wenn

$$\langle V \cap \mathbb{Z}^n \rangle_{\text{aff}} = V$$

gilt, wenn also V als affiner Raum von den ganzzahligen Vektoren in V erzeugt wird. Anderenfalls nennt man V irrational.

Satz 5.2.6. (LOVÁSZ, 1989)

Sei $S \subset \mathbb{R}^n$. Dann ist S dann und nur dann eine MKGF-Menge, wenn eine der beiden folgenden Bedingungen erfüllt ist:

- (1) S ist ein Polyeder von der Form $S = P + L$, wobei P ein Polytop und L ein von ganzzahligen Vektoren erzeugter Vektorraum ist, wobei gilt, dass $\dim \langle P \rangle + \dim L = n$, $S^\circ \cap \mathbb{Z}^n = \emptyset$ und jede Seitenfläche von S enthält einen Gitterpunkt in ihrem relativen Inneren.
- (2) S ist eine irrationale affine Hyperebene in \mathbb{R}^n .

BEWEIS. [BCCZ09, Theorem 2] □

Bemerkung 5.2.7. Sei $S = P + L$ eine MKGF-Menge wie in Satz 5.2.6(1). Dann ist P ein beschränktes Polygon, da die Seitenflächen von S in ihrem relativen Inneren jeweils mindestens einen Gitterpunkt enthalten. Aus diesen lässt sich nun ein Fundamentalparallelotop eines Teilgitters von \mathbb{Z}^n ergänzen, dessen Projektion auf $\langle P \rangle$ dann, sollte P unbeschränkt sein, echt in P enthalten wäre. Das ist aber ein Widerspruch zur Gitterpunktfreiheit von S .

Wir kommen nun zum Hauptresultat dieses Abschnittes:

Proposition 5.2.8. Die Menge D aus Gleichung (5.1) ist zulässig.

BEWEIS. Sei $x \in \partial \mathcal{V}_1^{\geq 0}$ beliebig. Nach Lemma 3.1.2 gibt es dann ein $y \in \mathcal{V}_2^{\geq 0} \setminus \{0\}$ mit $xy^{tr} = 0$. Betrachte für $\varepsilon > 0$ die Menge

$$M_\varepsilon := \{v \in \mathcal{V}_2^{\geq 0} \setminus \{0\} \mid xv^{tr} \leq \varepsilon\}.$$

Zu zeigen ist nun, dass $M_\varepsilon \cap D \neq \emptyset$ gilt, dann folgt die Behauptung mit Lemma 3.1.8.

Angenommen, das wäre falsch. Da M_ε offenbar konvex ist, gibt es dann eine MKGF-Menge $S \subset \mathbb{R}^n$ mit $M_\varepsilon \subset S$. Diese MKGF-Menge kann sicherlich keine affine Hyperebene sein, also gilt $S = P + L$ für ein Polytop P und einen rationalen Teilraum L von \mathbb{R}^n . Da P aber ein beschränktes Polytop ist, kann nicht $y \in P$ gelten, also ist $y = p + \ell$ für ein $p \in P$, $\ell \in L \setminus \{0\}$. Aus

Dimensionsgründen sind p und ℓ eindeutig bestimmt. Für $\lambda > 0$ folgt dann $\lambda y = \lambda p + \lambda \ell$ was aber wegen der Beschränktheit von P für große λ nicht stimmen kann. Also folgt $y \in L$. Analoges gilt für die Vektoren $v \in \mathcal{V}_2^{>0}$ mit $xv^{tr} = \varepsilon$: jedes solche v ist von der Form $v = v_0 + y'$, wobei v_0 fest gewählt ist mit $xv_0^{tr} = \varepsilon$ und $y' \in \mathbb{R}^n$ geeignet mit $xy'^{tr} = 0$. Man kann nun offenbar $n - 1$ linear unabhängige y' finden, so dass $\lambda y' \in \mathcal{V}_2^{>0}$ für alle $\lambda > \lambda_0 \geq 0$, also ergibt sich hier mit dem gleichen Argument wie oben eine Zerlegung $v = v_0 + y'$ mit $v_0 \in P, y' \in L$, also gilt $S \supset P_\varepsilon(x)$. Nach Korollar 5.2.3 enthält aber $P_\varepsilon(x)$ einen Gitterpunkt $\neq 0$ und das ist ein Widerspruch. Daher muss $M_\varepsilon \cap D \neq 0$ gelten und die Behauptung folgt. \square

5.3 Kürzeste Vektoren und Automorphismen

Wir kommen nun zu einer Methode, um die Menge $M_D(x)$ für $x \in \mathcal{V}_1^{>0}$ zu berechnen. Im Wesentlichen verwendet das Verfahren auf einem Algorithmus aus [PZ89, S. 188-190] zur Berechnung aller kurzen Vektoren einer (positiv definiten) quadratischen Form Q bzw. eines Gitters L , d.h. aller Vektoren $x \in \mathbb{Z}^n$ mit $Q(x) \leq C$ bzw. $x \in L$ mit $N(x) \leq C$ für gegebenes $C > 0$. Dazu zunächst eine vorbereitende

Bemerkung 5.3.1. Sei $A \in \mathbb{Z}^{n \times n}$ hyperbolisch.

(i) Für $x \in \mathcal{V}_1^{>0}$ ist $-xA \in \mathcal{V}_2^{>0}$.

(ii) Gilt $x \in \mathbb{Z}^n$, so ist $\mu_D(x) \leq -xAx^{tr} =: N(x)$.

BEWEIS. (i) Man betrachte zunächst den Fall der Standardform $H_n = \text{diag}(-1, 1, \dots, 1)$ mit dem zugehörigen selbstdualen Kegel \mathcal{C} wie in Lemma 5.1.1.

Sei also $x = (x_1, \dots, x_n) \in \mathcal{C}$ beliebig. Dann gilt

$$(-xH_n)H_n^{-1}(-xH_n)^{tr} = xH_n^{tr}x^{tr} = xH_nx^{tr} < 0$$

und

$$(-xH_n)e_1^{tr} = (-x) \cdot (-e_1^{tr}) = x_1 > 0.$$

Also folgt $-xH_n \in \mathcal{C}$.

Sei nun $A \in \mathbb{Z}_{sym}^{n \times n}$ eine beliebige hyperbolische Matrix mit fest gewähltem $y \in \mathbb{Z}^n$ im Sinne von Lemma 5.1.2.

Für $x \in \mathcal{V}_1^{>0}$ gilt dann

$$(-xA)A^{-1}(-xA)^{tr} = xAx^{tr} < 0$$

und wegen $xT^{-1} \in \mathcal{C}$ ist nach obiger Rechnung $-xT^{-1}H_n \in \mathcal{C}$, also $-xT^{-1}H_nT^{-tr} = -xA$ in $\mathcal{C} \cdot T^{tr} = \mathcal{V}_2^{>0}$.

(ii) Folgt aus (i), denn demnach ist $-xA \in \mathcal{V}_1^{>0} \cap \mathbb{Z}^n \subseteq D$, also gilt

$$\mu_D(x) = \min\{xd^{tr} \mid d \in D\} \leq x(-xA)^{tr} = -xAx^{tr}. \quad \square$$

Lemma 5.3.2. *Sei $A \in \mathbb{Z}^{n \times n}$ eine hyperbolische Matrix mit zugehörigen dualen Kegeln $\mathcal{V}_1^{>0}$ und $\mathcal{V}_2^{>0}$ und D wie in Gleichung (5.1). Weiter sei für $\lambda \in \mathbb{R}$ und $x \in \mathcal{V}_1^{>0} \cap \mathbb{Z}^n$*

$$H_\lambda(x) := \{-\lambda xA + y \in \mathbb{R}^n \mid xy^{tr} = 0\}.$$

Ist $\lambda > 0$ minimal, so dass die (stets endliche) Menge

$$M_\lambda = \{d \in H_\lambda(x) \cap \mathbb{Z}^n \mid (d + \lambda xA)A^{-1}(d + \lambda xA)^{tr} \leq -\lambda^2 xAx^{tr} \text{ und } y_2 d^{tr} \geq 0\}$$

mit y_2 wie in Lemma 5.1.2 nicht leer ist, so gilt

$$M_\lambda = M_D(x).$$

BEWEIS. Sei $x \in \mathcal{V}_1^{>0} \cap \mathbb{Z}^n$, $\lambda > 0$ und $z = -\lambda xA + y \in H_\lambda(x)$. Dann gilt

$$\begin{aligned} xz^{tr} &= \lambda N(x) \quad \text{und} \\ zA^{-1}z^{tr} &= (-\lambda xA)A^{-1}(-\lambda xA)^{tr} - 2 \underbrace{(\lambda xA)A^{-1}y^{tr}}_{=\lambda xy^{tr}=0} + yA^{-1}y^{tr} \\ &= -\lambda^2 N(x) + yA^{-1}y^{tr}. \end{aligned}$$

Ist also λ minimal, so dass $d = -\lambda xA + y \in H_\lambda(x) \cap \mathbb{Z}^n$ existiert, so ist auch xd^{tr} minimal. Damit $d \in \mathcal{V}_1^{>0}$ gilt, muss zudem zumindest gelten, dass

$$dA^{-1}d^{tr} = -\lambda^2 N(x) + yA^{-1}y^{tr} \leq 0,$$

also

$$(d + \lambda xA)A^{-1}(d + \lambda xA)^{tr} \leq \lambda^2 N(x).$$

Damit gilt für jedes $\lambda > 0$, dass $M_\lambda \subset D$, also folgt $M_\lambda = M_D(x)$ für das kleinste λ mit $M_\lambda \neq \emptyset$.

Da xA bezüglich der von A^{-1} induzierten Bilinearform senkrecht auf $H_0(x)$ steht,

$$(xA)A^{-1}(-0 \cdot xA + y) = 0,$$

und $(-xA)A^{-1}(-xA)^{tr} < 0$, ist die Einschränkung von A^{-1} auf den Teilraum H_0 positiv definit, so dass die Menge aller $y \in H_0(x)$ mit $d = -\lambda xA + y \in \mathbb{Z}^n$ und

$$yA^{-1}y^{tr} \leq \lambda^2 N(x)$$

beschränkt und diskret in \mathbb{R}^n , also endlich ist. \square

Bemerkung 5.3.3. (i) *Es sei $z = -\lambda xA + y \in H_\lambda(x) \cap \mathbb{Z}^n$. Da nach Voraussetzung $x \in \mathbb{Z}^n$ und $A \in \mathbb{Z}^{n \times n}$ gilt, folgt*

$$zx^{tr} = -\lambda \underbrace{xAx^{tr}}_{\in \mathbb{Z}} + \underbrace{yx^{tr}}_{=0} \in \mathbb{Z},$$

also enthält $H_\lambda(x)$ höchstens dann ganzzahlige Punkte, wenn $\lambda \in \frac{1}{N(x)}\mathbb{Z}$, wobei wieder $N(x) := -xAx^{tr}$ sei.

(ii) Es gilt

$$H_1(x) = \{-xA + y \mid xy^{tr} = 0\}.$$

Diese Menge enthält offensichtlich mindestens einen ganzzahligen Punkt, nämlich $-xA$. Dieser erfüllt trivialerweise die Ungleichung

$$(-xA + xA)A^{-1}(-xA + xA)^{tr} = 0 \leq -xAx^{tr},$$

also kann man sich auf der Suche nach dem geeigneten minimalen λ aus Lemma 5.3.2 auf die Werte $\lambda = \frac{1}{N(x)}, \frac{2}{N(x)}, \dots, \frac{N(x)}{N(x)}$ beschränken, so dass man nach höchstens $N(x)$ -maliger Verwendung des erwähnten Algorithmus 2.3.4 die Menge $M_D(x)$ bestimmt hat.

Wir kommen nun zu einer Methode, um den Stabilisator eines D -perfekten Punktes auszurechnen.

Lemma 5.3.4. (i) Da die Matrix A stets ganzzahlig ist, werden alle D -perfekten Punkte rational sein. Man kann sich also bis auf Skalierung darauf beschränken, ganzzahlige D -perfekte Punkte zu betrachten.

(ii) Sei x ein ganzzahliger D -perfekter Punkt und $g \in S_x := \text{Stab}_{\text{Aut}_{\mathbb{Z}}(A)}(x)$. Dann induziert g einen Automorphismus des Gitters

$$L(x) := (xA)^\perp \cap \mathbb{Z}^n.$$

(iii) A eingeschränkt auf den von $L(x)$ erzeugten Teilraum von \mathbb{R}^n induziert ein positiv definites Skalarprodukt.

BEWEIS. (i) Klar.

(ii) Für $v_1, v_2 \in L(x)$ gilt

$$xA(v_i g)^{tr} = xg^{-1}gAg^{tr}v_i^{tr} = xAv_i = 0,$$

also gilt $v_i g \in L(x)$, und es ist

$$(v_1 g)A(v_2 g)^{tr} = v_1 g A g^{tr} v_2^{tr} = v_1 A v_2^{tr},$$

also induziert g einen Automorphismus von $L(x)$.

(iii) Der Vektor x steht bezüglich A senkrecht auf $L(x)$ und erfüllt, da er insbesondere aus $\mathcal{V}_1^{>0}$ stammt die Ungleichung

$$xAx^{tr} < 0.$$

Also ist A auf dem (bezüglich A) orthogonalen Komplement von x , das offenbar von $L(x)$ erzeugt wird, positiv definit. \square

Bemerkung 5.3.5. Mittels der letzten Bemerkung ist es nun möglich, den Stabilisator eines ganzzahligen D -perfekten Punktes x zu berechnen:

Sei B' eine Gitterbasis von $L(x)$, die durch x zu einer Basis B von \mathbb{R}^n ergänzt wird. Bezüglich dieser Basis setzen sich dann die Automorphismen

fort zu einer Gruppe G_x , die auf $L(x)$ als Automorphismen und auf x trivial operiert. Bezüglich der Basis B haben die Elemente von G_x also die Form

$$\left(\begin{array}{c|c} B'gB' & 0 \\ \hline 0 & 1 \end{array} \right), g \in \text{Aut}(L(x)).$$

Bezüglich der Standardbasis (bzw. irgendeiner Gitterbasis von \mathbb{Z}^n) haben die Matrizen in G_x i.A. echt-rationale Einträge, da das Gitter $L(x) \oplus \mathbb{Z}x$ ein Teilgitter vom Index $N(x) = -xAx^{tr}$ in \mathbb{Z}^n ist. Man kann nun den Stabilisator S_x von x als Untergruppe in G_x wiederfinden: G_x operiert auf der Menge der Teilgitter von \mathbb{R}^n und S_x ist genau der Stabilisator von \mathbb{Z}^n . Um algorithmisch die Gleichheit zweier Gitter zu testen, kann man die HERMITE-Normalform (vgl. Satz 2.1.7) ihrer Basismatrizen vergleichen.

Zum Abschluss dieses Abschnittes folgt noch eine Methode, mit deren Hilfe man für zwei ganzzahlige perfekte Punkte $x, y \in \mathcal{V}_1^{>0}$ ein Element $\omega \in \text{GL}_n(\mathbb{Z})$ mit $y\omega = x$ und $\omega A\omega^{tr} = A$ zu finden. Man beachte, dass für die Existenz eines solchen ω notwendigerweise $xAx^{tr} = yAy^{tr}$ gelten muss. Die wesentliche Idee dabei ist wieder, die eigentliche Berechnung auf positiv definite Teilgitter zurückzuziehen und das Ergebnis dann geeignet fortzusetzen:

Lemma 5.3.6. *Seien $x, y \in \mathcal{V}_1^{>0} \cap \mathbb{Z}^n$ D -perfekte Punkte mit $xAx^{tr} = yAy^{tr}$ und $L(x), L(y)$ wie in Lemma 5.3.4(ii) und $\alpha : L(x) \rightarrow L(y)$ eine ganzzahlige Isometrie von Gittern. Dann gilt:*

- (i) *Es existiert eine rationale Fortsetzung α' von α auf \mathbb{R}^n , die $L(x)$ auf $L(y)$ und x auf y abbildet.*
- (ii) *Jedes $\omega \in \Omega := \text{Aut}_{\mathbb{Z}}(A)$ mit $x\omega = y$ ist von der Form $s\alpha'_s \cdot h$ für α' wie in (i) und ein $h \in G_y$, wobei G_y wie in Bemerkung 5.3.5 definiert sei. Es bezeichne S die Standardbasis von \mathbb{R}^n .*

BEWEIS. (i) Da jede Gitterbasis C_x von $L(x)$ durch x und jede Gitterbasis C_y von $L(y)$ durch y zu einer Basis B_x bzw. B_y von \mathbb{R}^n ergänzt werden kann, ist die Fortsetzbarkeit klar.

Es gilt dann bezüglich dieser angepassten Basen

$${}_{B_x}\alpha'_{B_y} = \left(\begin{array}{c|c} C_x\alpha C_y & 0 \\ \hline 0 & 1 \end{array} \right).$$

Bezüglich der Standardbasis ist i.A. $s\alpha'_s$ natürlich nicht ganzzahlig.

(ii) Klar. Es reicht hierbei natürlich aus, h eine Transversale von $G_y/\text{Stab}_{\Omega}(y)$ durchlaufen zu lassen. \square

Bemerkung 5.3.7. *Es ist durchaus möglich, dass keine Fortsetzung einer Isometrie $L(x) \rightarrow L(y)$ auf \mathbb{R}^n ganzzahlig ist, wie das Beispiel 3 in Beispiel 6.1.1 zeigt.*

5.4 Reflektive Gruppen

Wie sich in Abschnitt 5.5 zeigen wird, ist es hilfreich, sich mit dem Konzept von reflektiven Gruppen zu beschäftigen. Ursprünglich geht dieses Konzept auf COXETER (vgl. [Cox34]) zurück und wurde in den späten 1960er Jahren von VINBERG wesentlich weiterentwickelt. Eine ausführlichere Zusammenfassung der erreichten Resultate, die in der folgenden Zeit u.a. von NIKULIN und VINBERG-KAPLINSKAYA findet sich jeweils in der Einleitung zu [All12] und [Mc11].

Die Darstellung hier stützt sich hauptsächlich auf [Vin72] und [Mc11].

Für den Rest dieses Abschnitts sei eine hyperbolische quadratische Form

$$f(x) = xAx^{tr}$$

für eine Matrix $A \in \mathbb{Z}_{sym}^{n \times n}$ mit Signatur $(n-1, -1)$ fixiert. Wir betrachten den zu A gehörigen Kegel $\mathcal{V}_1^{>0}$.

Definition 5.4.1. *Unter dem $n-1$ -dimensionalen LOBAČEVSKY-Raum \mathfrak{L}_f von f versteht man die Menge der Strahlen in $\mathcal{V}_1^{\geq 0}$,*

$$\mathfrak{L}_f = \mathcal{V}_1^{\geq 0} / \mathbb{R}_{\geq 0},$$

wobei die Strahlen in $\partial\mathcal{V}_1^{>0}$ als unendlich ferne Punkte zu identifizieren sind. Das invariante Maß auf \mathfrak{L}_f wird mit μ bezeichnet.

Eine k -dimensionale Ebene S in \mathfrak{L}_f ist die Menge der Strahlen im Schnitt eines $k+1$ -dimensionalen Teilraums \mathcal{U} von \mathbb{R}^n mit $\mathcal{V}_1^{\geq 0}$,

$$S = (\mathcal{U} \cap \mathcal{V}_1^{\geq 0}) / \mathbb{R}_{\geq 0}.$$

Man nennt \mathcal{U} dann die Fortsetzung von S , die mit $cn(S)$ bezeichnet wird.

Definition 5.4.2. (i) Sei H eine $(n-1)$ -dimensionale Hyperebene in \mathfrak{L}_f . Dann bezeichnen H^+ und H^- die durch H definierten abgeschlossenen Halbräume. Die Fortsetzung dieser Halbräume auf \mathbb{R}^n wird ebenfalls mit $cn(H^+)$ bzw. $cn(H^-)$ bezeichnet.

Für eine Familie $H_i, i \in I$ von Hyperbenen heißt der Schnitt

$$P = \bigcap_{i \in I} H_i$$

genau dann ein konvexes Polyeder in \mathfrak{L}_f , falls

- (1) jede (bezüglich der hyperbolischen Metrik) beschränkte Teilmenge von \mathfrak{L}_f wird nur von endlich vielen Hyperbenen H_i geschnitten,
- (2) P enthält eine nicht-leere offene Menge.

(ii) Die Fortsetzung eines konvexen Polyeders P wie in (i) ist definiert als

$$cn(P) = \bigcap_{i \in I} cn(H_i^-)$$

- (iii) Man nennt ein Polyeder endlich, falls es ein Schnitt von endlich viele Halbräumen ist.
- (iv) Ein Polyeder heißt beschränkt, falls es keine unendlich fernen Punkte in \mathfrak{L}_f enthält.

Bemerkung 5.4.3. Sei $P = \bigcap_{i \in I} H_i$ ein Polyeder in \mathfrak{L}_f .

- (i) Ist P beschränkt, so ist P endlich (nach Definition).
- (ii) Das Maß von P ist genau dann endlich, wenn seine Fortsetzung im Abschluss des Kegels liegt,

$$\mu(P) < \infty \quad \Leftrightarrow \quad cn(P) \subset \mathcal{V}_1^{\geq 0}.$$

Satz 5.4.4. (VINBERG, 1972)

Sei Ω die Automorphismengruppe eines hyperbolischen Gitters zur Form f , Γ die von allen Spiegelungen in Ω erzeugte Untergruppe mit Fundamentalbereich P (aufgefasst als Polyeder in \mathfrak{L}_f). Dann gibt es eine Untergruppe $H \leq \text{Sym}(P)$ mit

$$\Omega = \Gamma \rtimes H.$$

Hierbei ist $\text{Sym}(P)$ die Symmetriegruppe von P .

BEWEIS. Offenbar ist $\Gamma \trianglelefteq \Omega$, da für eine Spiegelung $\gamma \in \Gamma$ an der Hyperebene H und ein beliebiges $\omega \in \Omega$ das Element γ^ω seinerseits eine Spiegelung an der Hyperebene $H\omega$, also in Γ enthalten. Daruas folgt auch, dass die Menge aller Spiegelungshyperebenen von Γ unter Ω invariant bleibt.

Sei nun $\omega \in \Omega$ beliebig. Dann ist das Polyeder $P\omega$ ebenfalls ein Fundamentalbereich für Γ , so dass ein $\gamma \in \Gamma$ existieren muss mit $P\omega = P\gamma$. Demnach gilt

$$\omega = \gamma \cdot \eta, \quad \text{für ein } \gamma \in \Gamma, \eta \in \text{Sym}(P).$$

Nun ist $\Gamma \cap \text{Sym}(P)$ offenbar trivial, so dass obige Zerlegung eindeutig ist, was die Behauptung zeigt. \square

5.5 Endlichkeit des Restklassengraphen

Bis hierher haben wir konstruktive Methoden vorgestellt, die Automorphismengruppe eines hyperbolischen Gitters mit GRAM-Matrix A zu berechnen. Bisher ist aber noch nicht klar, dass dieses Verfahren immer terminiert, was äquivalent dazu ist, dass der Restklassengraph $\Gamma_D / \text{Aut}_{\mathbb{Z}}(A)$ stets endlich ist. Allerdings zeigen die Resultate aus Abschnitt 5.1, dass zumindest die Ausgabe des Algorithmus korrekt ist, sofern er terminiert.

Eine hinreichende, aber nicht notwendige Bedingung (vgl. Kapitel 6) für das Terminieren des Algorithmus mit den Begriffen aus Abschnitt 5.4 lautet wie folgt.

Proposition 5.5.1. Sei $A \in \mathbb{Z}_{hyp}^{n \times n}$ eine ganzzahlige hyperbolische Matrix und Ω seine Automorphismengruppe. Dann terminiert Algorithmus 3.2.11 für die Bestimmung von Ω , falls Ω^{ad} reflektiv ist.

BEWEIS. Sei Ω^{ad} reflektiv. Dann existiert eine Spiegelungsgruppe $\Gamma \trianglelefteq \Omega^{ad}$ mit $[\Omega^{ad} : \Gamma] < \infty$. Ein Fundamentalbereich \mathbb{F}_Γ von Γ in $\mathcal{V}_2^{>0}$ ist dann eine Pyramide mit endlich vielen Seitenflächen, also gibt es, indem man zur dualen Beschreibung des Fundamentalbereiches übergeht, endlich viele Vektoren v_1, \dots, v_ℓ mit

$$\mathbb{F}_\Gamma = \left\{ \sum_{i=1}^{\ell} a_i v_i \mid a_i > 0 \right\}.$$

Da Γ eine Untergruppe von Ω^{ad} ist, enthält \mathbb{F}_Γ nach einen Fundamentalbereich von Ω^{ad} in $\mathcal{V}_2^{>0}$. Mit Proposition 3.2.15 ergibt sich dann aber, dass es nur endlich viele D -perfekte Punkte modulo Ω gibt, was äquivalent dazu ist, dass Algorithmus 3.2.11 terminiert. \square

Korollar 5.5.2. (i) Ist $A = gH_n g^{tr}$ mit $H_n = \text{diag}(-1, 1, \dots, 1)$ und $g \in \text{GL}_n(\mathbb{Z})$, dann terminiert der Algorithmus für $n \leq 19$.

(ii) Ist $A = gH_n^{(2)} g^{tr}$ mit $H_n^{(2)} = \text{diag}(-2, 1, \dots, 1)$ und $g \in \text{GL}_n(\mathbb{Z})$, dann terminiert der Algorithmus für $n \leq 14$.

(iii) Ist $A = gH_n^{(3)} g^{tr}$ mit $H_n^{(3)} = \text{diag}(-3, 1, \dots, 1)$ und $g \in \text{GL}_n(\mathbb{Z})$, dann terminiert der Algorithmus für $n \leq 13$.

BEWEIS. (i) und (ii): Nach [Vin72, §4] sind die entsprechenden Formen genau in den angegebenen Dimensionen reflektiv.

(iii): In [Mc11] wird gezeigt, dass die Form zu $H_n^{(3)}$ für $n \leq 13$ reflektiv ist und sonst nicht. \square

Zum Abschluss dieses Kapitels sei noch erwähnt, dass sich die hier entwickelten Methoden nicht ohne Weiteres auf die Berechnung von Automorphismen mehrfach indefiniter Gitter, d.h. Gitter, deren GRAM-Matrix eine Signatur $(p, -q)$ mit $p \geq q \geq 2$ und $p + q = n$ hat, anwenden lassen, obgleich die generelle Idee für die Berechnung kurzer Vektoren und Stabilisatoren aus Abschnitt 5.3, die Berechnung auf besser handhabbare Gitter zurückzuziehen (z.B. von hyperbolischen auf EUKLIDISCHE Gitter) dies nahelegt.

Bemerkung 5.5.3. Besitzt eine Gruppe G einen Fundamentalbereich \mathbb{F} im Sinne von Definition 3.3.1, der nur endlich viele Nachbarn besitzt, so ist für jedes $x \in \mathfrak{X}$ die Ordnung des Stabilisators $\text{Stab}_G(x)$ beschränkt. Gibt es also Punkte, deren Stabilisator nicht endlich ist, dann kann der Fundamentalbereich nicht nur endlich viele Nachbarn besitzen.

BEWEIS. Sei $x \in \mathfrak{X}$ und $g \in G$ mit $xg \in \mathbb{F}$. Wegen $\text{Stab}_G(x) = g \text{Stab}_G(xg)g^{-1}$ kann man sich für die Untersuchung der Stabilisatoren also auf Elemente $x \in \mathbb{F}$ einschränken.

Sei nun

$$G_{\mathbb{F}} := \{h \in G \mid \mathbb{F}h \cap \mathbb{F} \neq \emptyset\}$$

Nach Voraussetzung ist $G_{\mathbb{F}}$ endlich und offensichtlich gilt für $x \in \mathbb{F}$, dass $\text{Stab}_G(x) \subseteq G_{\mathbb{F}}$. Damit ist die Ordnung jedes Stabilisators durch $|G_{\mathbb{F}}|$ beschränkt. \square

5.6 Hyperbolische Gitter und der WATSON-Prozess

Wie in Abschnitt 2.2 gesehen kann der WATSON-Prozess die Diskriminante eines hyperbolischen Gitters (L, Φ) reduzieren. Ferner enthält die Automorphismengruppe von $\text{Watson}(L)$ die Automorphismengruppe von L als Untergruppe von endlichem Index (vgl. Bemerkung 2.2.2). Diese Beobachtung kann man sich zu Nutze machen um die Berechnung der Automorphismengruppe von L zu beschleunigen:

Bemerkung 5.6.1. *Sei A die GRAM-Matrix eines hyperbolischen Gitters (L, Φ) bezüglich einer fest gewählten Basis B . Es bezeichne $\text{Watson}(A)$ die GRAM-Matrix von $\text{Watson}(L)$ bezüglich einer Basis B' von $\text{Watson}(L)$. Dann ist $\text{Aut}_{\mathbb{Z}}(\text{Watson}(A))$ die Darstellung von $\text{Aut}(\text{Watson}(L))$ bezüglich der Basis B' . Stellt man diese wieder in der Basis B dar, so ergibt sich eine Untergruppe von $\text{GL}_n(\mathbb{Q})$, die $\text{Aut}_{\mathbb{Z}}(A)$ enthält. Man kann dann $\text{Aut}_{\mathbb{Z}}(A)$ genau wie in Bemerkung 5.3.5 beschrieben mittels Bahnenalgorithmus bestimmen.*

Alle bisher betrachteten Beispiele zeigen, dass sich auf diese Weise die Berechnung von $\text{Aut}(L)$ teilweise extrem beschleunigen lässt (vgl. Beispiel 6.1.2).

Dies hat mehrere Gründe: Da sich die Automorphismengruppe von $\text{Watson}(L)$ die von L enthält, kann man erwarten, dass es für $\text{Watson}(L)$ eher weniger Äquivalenzklassen D -perfekter Punkte gibt als für L . Außerdem hängt die Laufzeit des Verfahrens zur Berechnung der D -minimaler Vektoren eines Vektors $x \in \mathcal{V}_1^{>0}$ aus Lemma 5.3.2 im Wesentlichen quadratisch von seiner Norm $N(x) = -xAx^{tr}$ als auch von der Diskriminante von L ab. Verkleinert man letztere, so ist damit zu rechnen, dass die Berechnung von $M_D(x)$ sich beschleunigt.

Kapitel 6

Beispiele

Alle Rechnungen, deren Ergebnisse hier präsentiert werden, wurden auf einem Quad-Core AMD Opteron(tm) Prozessor mit 2.3 GHz mit der Magma-Version 2.18-9 durchgeführt. Die angegebenen Rechenzeiten wurde durch den `time`-Befehl in Magma ermittelt.

6.1 Die Automorphismengruppen zu einigen hyperbolischen Gittern

In diesem Abschnitt werden auch die D -VORONOI-Graphen für einige Beispiele angegeben. Die nicht-indizierten schwarzen Kanten repräsentieren dabei direkte Nachbarn, blaue Kanten eine Verbindung, die durch ein nicht-triviales verbindendes Element bewirkt wird, das jeweils durch den Index angegeben wird.

Soweit möglich werden sowohl die benötigten Rechenzeiten unter Verwendung der Methode aus Bemerkung 3.1.17 und der aus dem Beweis zu Proposition 3.1.11 zur Berechnung des VORONOI-Nachbarn angegeben (vgl. Appendix B.2).

Für einige dreidimensionale Beispiele wird zudem ein Querschnitt der Vereinigung aller D -VORONOI-Bereiche zu den berechneten D -perfekten Punkten abgebildet, d.h. die Projektion auf eine Ebene senkrecht zur Kegelachse des Kegels $\mathcal{V}_2^{\geq 0}$. Die Graphiken hierzu wurden mit dem Software-Paketes Sage ([Sag]) erzeugt.

6.1.1 Standardform

Mit dem implementierten Verfahren lässt sich die Automorphismengruppe für das Standardgitter mit GRAM-Matrix

$$H = \begin{pmatrix} -1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{pmatrix}$$

bis einschließlich Dimension 7 mit vertretbarem Aufwand berechnen. Die Berechnung in Dimension 8 ist, wie beim klassischen VORONOI-Algorithmus auch,

zu aufwändig, um sie ohne Weiteres durchführen zu können.

Dimension 2

Hier gibt es nur einen D -perfekten Punkt

$$x_1 = (1 \ 0).$$

Dieser besitzt keine Richtungen, also ist die Automorphismengruppe in diesem Fall endlich und besteht nur aus dem Stabilisator

$$\text{Stab}(x_1) = \left\langle \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \right\rangle \cong C_2$$

von x_1 und der VORONOI-Graph¹ besitzt nur einen einzigen Knoten und keine Kanten.

$$\bullet_{x_1}$$

Die Berechnung dauerte sehr viel weniger als 1 Sekunde.

Dimension 3

Auch hier gibt es bis auf Operation der Automorphismengruppe einen D -perfekten Punkt

$$x_1 = (1 \ 0 \ 0).$$

Dieser besitzt nun 4 Richtungen. Jeder Nachbar ist jedoch unter der Operation äquivalent zu x_1 , so dass die Automorphismengruppe außer vom Stabilisator

$$\text{Stab}(x_1) = \left\langle \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix} \right\rangle \cong C_2^3$$

auch von dem verbindenden Element

$$c_{1,1} = \begin{pmatrix} 3 & 2 & 2 \\ 2 & 1 & -2 \\ 2 & 2 & -1 \end{pmatrix}$$

erzeugt wird. Als D -VORONOI-Graph erhält man so

$$c_{1,1} \circ \bullet_{x_1}.$$

Insgesamt benötigte auch diese Rechnung sehr viel weniger als eine Sekunde.

Der D -VORONOI-Bereich des perfekten Punktes ist eine Pyramide mit quadratischer Grundfläche, deren Seitenkanten auf dem Rand des selbstdualen Kegels

$$\mathcal{C} = \{(x_1, x_2, x_3) \in \mathbb{R}^3 \mid x_2^2 + x_3^2 < x_1^2 \text{ und } x_1 > 0\}$$

¹die Graphen in diesem Kapitel wurden mit dem L^AT_EX-Paket xy-pic erzeugt

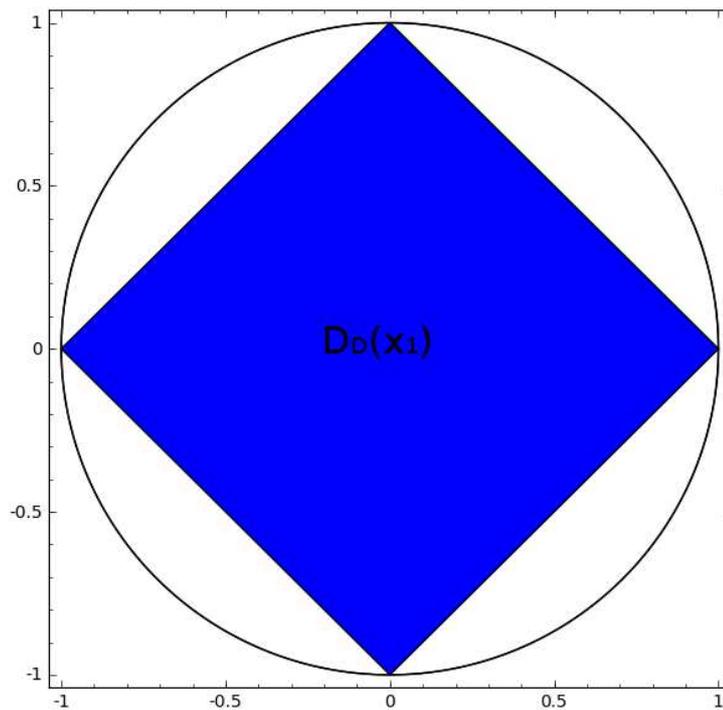


Abbildung 6.1: Fundamentalbereich für die Form $H_3 = \text{diag}(-1, 1, 1)$.

liegen. Die Kegelachse ist hier die x_1 -Achse. Schneidet man diesen Kegel und die enthaltene Pyramide mit der Ebene $x_1 = 1$, so ergibt sich das untenstehende Schnittbild. Das blaue Quadrat ist hierbei der Schnitt des D -VORONOI-Bereiches von x_1 mit der Ebene, der Kreis beschreibt entsprechend den Rand des Kegels.

Dimension 4

Der einzige perfekte Punkt

$$x_1 = (1 \ 0 \ 0 \ 0)$$

besitzt hier 8 Richtungen, wobei auch hier die entsprechenden Nachbarn zu x_1 äquivalent sind. Die Situation ist also genau wie in Dimension 3. Der Stabilisator hat 48 Elemente und ist gegeben durch

$$\text{Stab}(x_1) = \left\langle \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \end{pmatrix}, \right. \\ \left. \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \right\rangle \cong C_2 \times S_4$$

und erzeugt zusammen mit dem verbindenden Element

$$c_{1,1} = \begin{pmatrix} 2 & -1 & 1 & -1 \\ 1 & -1 & 0 & -1 \\ 1 & -1 & 1 & 0 \\ 1 & 0 & 1 & -1 \end{pmatrix}$$

die Automorphismengruppe. Der D -VORONOI-Graph ist wieder

$$c_{1,1} \circ x_1,$$

was man wieder in deutlich unter einer Sekunde erhält.

Dimension 5

Hier wird die Situation das erste Mal etwas komplexer, es gibt nämlich modulo $\text{Aut}_{\mathbb{Z}}(H)$ 2 perfekte Punkte

$$\begin{aligned} x_1 &= (1 \ 0 \ 0 \ 0 \ 0) \\ x_2 &= (3 \ -1 \ 1 \ -1 \ 1) \end{aligned}$$

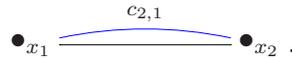
mit jeweils 16 bzw. 5 Richtungen. Die Stabilisatoren sind gegeben durch

$$\begin{aligned} \text{Stab}(x_1) &= \left\langle \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 \end{pmatrix}, \right. \\ &\quad \left. \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \right. \\ &\quad \left. \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix} \right\rangle \cong (((C_2 \times D_8) : C_2) : C_3) : C_2 : C_2 \\ \text{Stab}(x_2) &= \left\langle \begin{pmatrix} 2 & 0 & 1 & -1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 & -1 \\ 1 & 0 & 0 & -1 & 1 \\ -1 & 0 & -1 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 \end{pmatrix}, \right. \\ &\quad \left. \begin{pmatrix} 2 & -1 & 1 & -1 & 0 \\ 1 & -1 & 0 & -1 & 0 \\ -1 & 0 & -1 & 1 & 0 \\ 1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \right\rangle \cong S_5 \end{aligned}$$

Die Stabilisatoren haben 384^2 und 120 Elemente. Zusätzlich gibt es folgendes verbindendes Element:

$$c_{2,1} = \begin{pmatrix} 2 & -1 & 1 & -1 & 0 \\ 1 & -1 & 0 & -1 & 0 \\ 1 & -1 & 1 & 0 & 0 \\ 1 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

so dass der D -VORONOI-Graph folgende Gestalt hat.



Magma berechnete diese Informationen innerhalb von etwa 2 Sekunden.

Dimension 6

Auch hier gibt es wieder 2 D -perfekte Punkte

$$x_1 = (1 \ 0 \ 0 \ 0 \ 0 \ 0), x_2 = (3 \ -1 \ 1 \ 1 \ -1 \ -1).$$

Hierbei besitzen beide Punkte insgesamt 32 Richtungen. Ihre Stabilisatoren sind gegeben durch

$$\begin{aligned} \text{Stab}(x_1) = & \left\langle \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \end{pmatrix}, \right. \\ & \left. \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \right\rangle \\ \text{Stab}(x_2) = & \left\langle \begin{pmatrix} 2 & 0 & 1 & -1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 & -1 \\ 1 & 0 & 0 & -1 & 1 \\ -1 & 0 & -1 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 \end{pmatrix}, \right. \\ & \left. \begin{pmatrix} 2 & -1 & 1 & -1 & 0 \\ 1 & -1 & 0 & -1 & 0 \\ -1 & 0 & -1 & 1 & 0 \\ 1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \right\rangle, \end{aligned}$$

²die Isomorphietypen der Stabilisatoren wurden mit Hilfe von GAP 4 ([GAP]) bestimmt

wobei die Stabilisatoren jeweils 3840 bzw. 120 Elemente haben. Zusätzlich haben wir das verbindende Element

$$c_{2,2} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}.$$

Als D -VORONOI-Graphen erhält man nach ca. 70 Sekunden Rechenzeit:

$$\bullet_{x_1} \text{ ————— } \bullet_{x_2} \text{) } c_{2,2}.$$

Dimension 7

$$x_1 = (1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0),$$

$$x_2 = (3 \ 1 \ 1 \ -1 \ -1 \ 1 \ 1)$$

Diese Vektoren besitzen 64 bzw. 99 Richtungen und die Stabilisatoren mit jeweils 46080 bzw. 51840 Elementen haben folgende Erzeugendensysteme.

$$\text{Stab}(x_1) = \left\langle \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 \end{pmatrix}, \right.$$

$$\left. \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \right\rangle$$

$$\text{Stab}(x_2) = \left\langle \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 3 & 1 & 1 & 0 & -1 & 2 & 1 \\ -1 & 0 & 0 & 0 & 1 & -1 & 0 \\ -2 & -1 & -1 & 0 & 1 & -1 & -1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & -1 & -1 \\ -1 & 0 & -1 & 0 & 0 & -1 & 0 \end{pmatrix}, \begin{pmatrix} 3 & 0 & 1 & -1 & -1 & 1 & 2 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & -1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ -1 & 0 & 0 & 0 & 1 & 0 & -1 \\ -2 & 0 & -1 & 1 & 1 & -1 & -1 \end{pmatrix}, \right.$$

$$\left. \begin{pmatrix} 3 & 1 & 1 & -1 & 0 & 2 & 1 \\ -1 & 0 & 0 & 0 & 0 & -1 & -1 \\ -2 & -1 & -1 & 1 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 & 0 & -1 & 0 \\ -1 & -1 & 0 & 0 & 0 & -1 & 0 \end{pmatrix}, \begin{pmatrix} 2 & 1 & 1 & 0 & -1 & 0 & 0 \\ -1 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \right\rangle.$$

Mit den beiden verbindenden Elementen

$$c_{2,2} = \begin{pmatrix} 6 & 2 & 4 & -1 & -3 & 1 & 2 \\ -4 & -1 & -3 & 1 & 2 & -1 & -1 \\ -3 & -1 & -2 & 0 & 2 & 0 & -1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 1 & 0 & -1 & 1 & 1 \\ -1 & 0 & -1 & 0 & 0 & 0 & -1 \\ -2 & -1 & -1 & 1 & 1 & 0 & -1 \end{pmatrix}, \quad c_{2,1} = \begin{pmatrix} 4 & 2 & 1 & -1 & -2 & 2 & 1 \\ 1 & 0 & 0 & 0 & -1 & 1 & 0 \\ 1 & 1 & 0 & 0 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 2 & 1 & 0 & -1 & -1 & 1 & 1 \\ 2 & 1 & 1 & -1 & -1 & 1 & 0 \\ 2 & 1 & 1 & 0 & -1 & 1 & 1 \end{pmatrix}$$

ist nach 1450 Sekunden Rechenzeit ein vollständiges Erzeugendensystem der Automorphismengruppe gefunden. Der D -VORONOI-Graph hat die folgende Gestalt.

$$\bullet_{x_1} \text{ ————— } \bullet_{x_2} \text{) } c_{2,2}.$$

Insgesamt beanspruchten diese Berechnungen ca. 5.8 GB Arbeitsspeicher.

6.1.2 Andere Formen

Beispiel 6.1.1. (1) Dieses Beispiel diskutiert auch OPGENORTH im unveröffentlichten Teil von [Opg01].

Sei

$$A := \begin{pmatrix} 2 & 3 & 1 \\ 3 & 3 & 1 \\ 1 & 1 & 4 \end{pmatrix}, \quad A^{-1} = \frac{1}{11} \begin{pmatrix} -11 & 11 & 0 \\ 11 & -7 & -1 \\ 0 & -1 & 3 \end{pmatrix}.$$

Für dieses Beispiel gibt es insgesamt 4 perfekte Punkte

$$\begin{aligned} x_1 &= (-1 \ 1 \ 0), & x_2 &= (-3 \ 3 \ 1), \\ x_3 &= (-4 \ 3 \ 2), & x_4 &= (-9 \ 11 \ 3). \end{aligned}$$

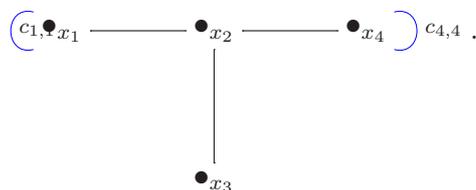
Hierbei haben x_1 und x_3 jeweils 4, x_2 und x_4 jeweils 3 Nachbarn. Die genauen Nachbarschaftsbeziehungen gibt der untenstehende Graph an. Der Stabilisator von x_2 ist trivial und es gilt

$$\begin{aligned} \text{Stab}(x_1) &= \left\langle \begin{pmatrix} 1 & -2 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \right\rangle \cong C_2 \\ \text{Stab}(x_3) &= \left\langle \begin{pmatrix} 23 & -18 & -12 \\ -8 & 5 & 4 \\ 56 & -42 & -29 \end{pmatrix}, \begin{pmatrix} 23 & -18 & -12 \\ 0 & 1 & 0 \\ 44 & -36 & -23 \end{pmatrix} \right\rangle \cong C_2 \times C_2, \\ \text{Stab}(x_4) &= \left\langle \begin{pmatrix} 5 & -8 & 0 \\ 3 & -5 & 0 \\ 1 & -2 & 1 \end{pmatrix} \right\rangle \cong C_2. \end{aligned}$$

Die verbindenden Elemente sind

$$c_{1,1} = \begin{pmatrix} 5 & -2 & 0 \\ 3 & -1 & 0 \\ 1 & 0 & -1 \end{pmatrix}, \quad c_{4,4} = \begin{pmatrix} 49 & -60 & -20 \\ 25 & -31 & -10 \\ 45 & -54 & -19 \end{pmatrix}.$$

Man erhält folgenden VORONOI-Graphen:



Die für die Berechnung benötigte Zeit betrug mit beiden Methoden weniger als eine Sekunde.

(2) Für

$$A = \begin{pmatrix} -1 & -3 & -1 \\ -3 & 14 & 8 \\ -1 & 8 & 11 \end{pmatrix}, \quad A^{-1} = \frac{1}{155} \begin{pmatrix} -90 & -25 & 10 \\ -25 & 12 & -11 \\ 10 & -11 & 23 \end{pmatrix}$$

gibt es folgende perfekte Punkte:

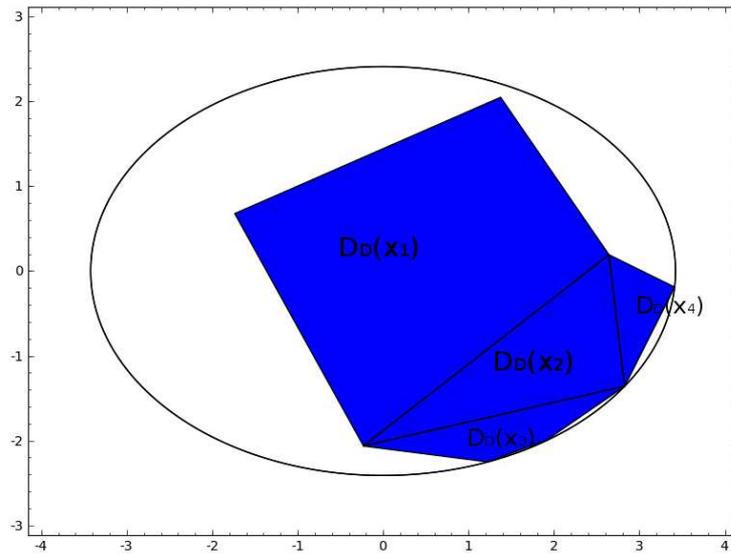


Abbildung 6.2: D -VORONOI-Bereiche der perfekten Punkte

$$\begin{array}{lll}
 x_1 = (1 & 0 & 0) & x_2 = (2 & 1 & -1) & x_3 = (2 & 1 & 0) \\
 x_4 = (9 & 0 & -2) & x_5 = (5 & 3 & -3) & x_6 = (12 & 5 & -7) \\
 x_7 = (3 & 2 & -1) & x_8 = (14 & 9 & -2) & x_9 = (21 & 8 & -12)
 \end{array}$$

Diese Punkte haben jeweils (in der aufgelisteten Reihenfolge) 8, 4, 6, 8, 4, 3, 4, 3 und 6 Nachbarn.

Die Stabilisatoren von x_2, x_5, x_6, x_7 und x_8 sind hierbei trivial, die übrigen sind zyklisch von Ordnung 2 und besitzen folgende Erzeuger:

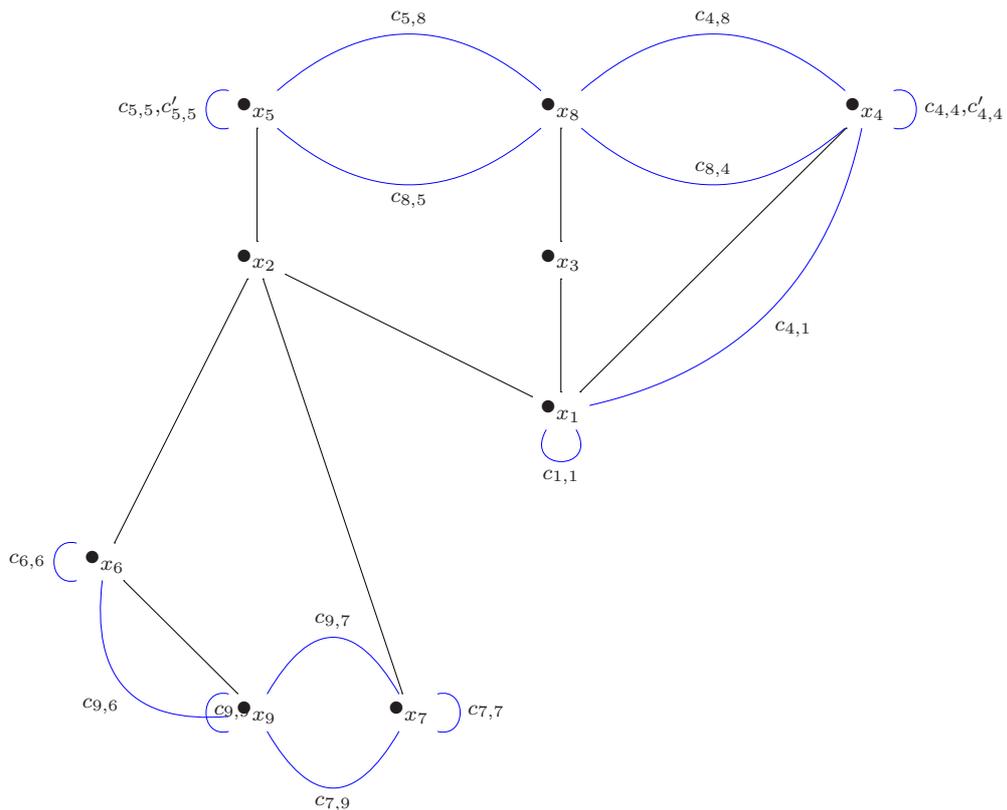
$$\begin{array}{ll}
 \text{Stab}(x_1) = \left\langle \begin{pmatrix} 1 & 0 & 0 \\ 6 & -1 & 0 \\ 2 & 0 & -1 \end{pmatrix} \right\rangle & \text{Stab}(x_3) = \left\langle \begin{pmatrix} 9 & 5 & 0 \\ -16 & -9 & 0 \\ -12 & -6 & -1 \end{pmatrix} \right\rangle \\
 \text{Stab}(x_4) = \left\langle \begin{pmatrix} 125 & 0 & -28 \\ 774 & -1 & -172 \\ 558 & 0 & -125 \end{pmatrix} \right\rangle & \text{Stab}(x_9) = \left\langle \begin{pmatrix} 1385 & 528 & -792 \\ 1974 & 751 & -1128 \\ 3738 & 1424 & -2137 \end{pmatrix} \right\rangle
 \end{array}$$

Zudem gibt es insgesamt 16 verbindende Elemente

$$\begin{array}{ll}
 c_{1,1} = \begin{pmatrix} 31 & -5 & 10 \\ 96 & -15 & 32 \\ -48 & 8 & -15 \end{pmatrix} & c_{4,1} = \begin{pmatrix} 125 & 0 & -28 \\ -24 & 1 & 4 \\ -308 & 0 & 69 \end{pmatrix} \\
 c_{4,4} = \begin{pmatrix} 561 & -28 & -84 \\ 3920 & -195 & -588 \\ 2440 & -122 & -365 \end{pmatrix} & c'_{4,4} = \begin{pmatrix} 145 & 9 & -44 \\ 966 & 59 & -292 \\ 642 & 40 & -195 \end{pmatrix} \\
 c_{4,8} = \begin{pmatrix} 1005 & 5 & -232 \\ -1784 & -9 & 412 \\ -1048 & -6 & 243 \end{pmatrix} & c_{5,5} = \begin{pmatrix} 289 & 180 & -180 \\ -96 & -59 & 60 \\ 368 & 230 & -229 \end{pmatrix}
 \end{array}$$

$$\begin{aligned}
 c'_{5,5} &= \begin{pmatrix} 51 & 35 & -30 \\ -40 & -27 & 24 \\ 40 & 28 & -23 \end{pmatrix} & c_{5,8} &= \begin{pmatrix} 109 & 57 & -68 \\ -186 & -97 & 116 \\ -82 & -42 & 51 \end{pmatrix} \\
 c_{6,6} &= \begin{pmatrix} 139 & 63 & -84 \\ 120 & 53 & -72 \\ 320 & 144 & -193 \end{pmatrix} & c_{7,9} &= \begin{pmatrix} 591 & 432 & -224 \\ 880 & 643 & -332 \\ 1620 & 1184 & -613 \end{pmatrix} \\
 c_{7,7} &= \begin{pmatrix} 45 & 33 & -22 \\ -56 & -41 & 28 \\ 8 & 6 & -3 \end{pmatrix} & c_{8,5} &= \begin{pmatrix} 75 & 51 & -16 \\ 26 & 17 & -4 \\ 142 & 96 & -29 \end{pmatrix} \\
 c_{8,4} &= \begin{pmatrix} 9 & 5 & 0 \\ 70 & 39 & 0 \\ 30 & 16 & 1 \end{pmatrix} & c_{9,7} &= \begin{pmatrix} 231 & 96 & -136 \\ -374 & -155 & 220 \\ -118 & -48 & 69 \end{pmatrix} \\
 c_{9,9} &= \begin{pmatrix} 1575 & 603 & -902 \\ 2506 & 961 & -1436 \\ 4422 & 1694 & -2533 \end{pmatrix} & c_{9,6} &= \begin{pmatrix} 1385 & 528 & -792 \\ 1974 & 751 & -1128 \\ 3738 & 1424 & -2137 \end{pmatrix}
 \end{aligned}$$

Der D -VORONOI-Graph hat folgende Gestalt;



die Berechnungen hierzu dauerten mit der zweiten Methode ca. 6 Sekunden, mit der ersten ca. 40 Sekunden.

(3) Sei

$$A := \begin{pmatrix} -14 & -4 & -6 \\ -4 & 19 & -4 \\ -6 & -4 & 8 \end{pmatrix}, \quad A^{-1} = \frac{1}{1454} \begin{pmatrix} -68 & -28 & -65 \\ -28 & 74 & 16 \\ -65 & 16 & 141 \end{pmatrix}.$$

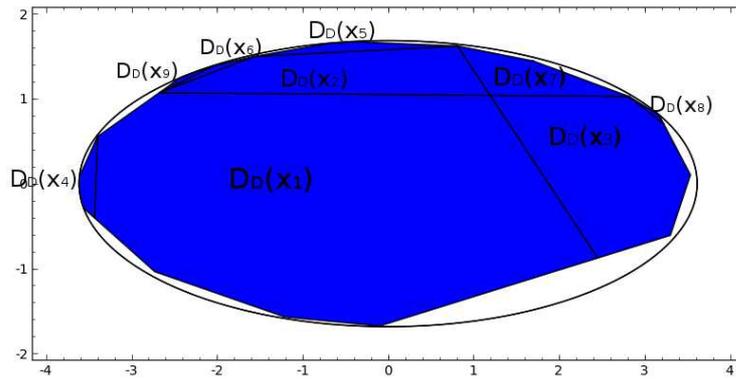


Abbildung 6.3: D -VORONOI-Bereiche der perfekten Punkte

Dann sind die Punkte

$$x = (125 \ 141 \ 7) \text{ und } y = (11 \ -7 \ 11)$$

benachbarte D -perfekte Punkte, für die zwischen den Gittern $L(x)$ und $L(y)$ eine ganzzahlige Isometrie existiert, die aber keine ganzzahlige Fortsetzung besitzt.

Insgesamt gibt es modulo der Automorphismengruppe 106 D -perfekte Punkte mit D -Minimum 1. Die zweite Methode lieferte dieses Ergebnis nach ca. 15:45 Stunden, die Berechnungen mit der ersten dauerten ca. 17:15 Stunden.

Dieses Beispiel zeigt zudem, dass der Algorithmus auch für nicht reflektive Gitter terminieren kann (vgl. Proposition 5.5.1), denn gemäß [All12, S. 24] ist der größte Primteiler der Diskriminante eines 3-dimensionalen hyperbolischen Gitters gerade 97, die Determinante von A ist aber $-2908 = -2^2 \cdot 727$, so dass dieses Gitter nicht reflektiv sein kann.

(4) Sei

$$A = \begin{pmatrix} 2 & 1 & 1 & 0 \\ 1 & 2 & 1 & 0 \\ 1 & 1 & 3 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}, \quad A^{-1} = \frac{1}{7} \begin{pmatrix} 5 & -2 & -1 & 0 \\ -2 & 5 & -1 & 0 \\ -1 & -1 & 3 & 0 \\ 0 & 0 & 0 & -7 \end{pmatrix}$$

Dann gibt es genau einen perfekten Punkt

$$x_1 = (0 \ 0 \ 0 \ 1),$$

der insgesamt 12 Richtungen hat. Zusätzlich zum Stabilisator

$$\text{Stab}(x_1) = \left\langle \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} -1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \right\rangle$$

von Ordnung 12 gibt es 2 verbindende Elemente

$$c_{1,1} = \begin{pmatrix} 1 & 1 & 0 & -2 \\ 0 & 3 & 0 & -4 \\ 0 & 1 & 1 & -2 \\ 0 & -2 & 0 & 3 \end{pmatrix},$$

$$c'_{1,1} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 2 & 0 & -1 & -2 \\ -1 & 0 & 1 & 2 \end{pmatrix},$$

die insgesamt die Automorphismengruppe erzeugen. Der Restklassengraph besitzt demnach genau einen Knoten mit zwei Schleifen.

$$c_{1,1} \left(\bullet_{x_1} \right) c'_{1,1}.$$

Beispiel 6.1.2. (i) Die quadratische Form definiert durch die Matrix

$$A = \begin{pmatrix} 9 & -2 & 7 \\ -2 & -2 & 8 \\ 7 & 8 & 9 \end{pmatrix}, \quad A^{-1} = \frac{1}{450} \begin{pmatrix} 41 & -37 & 1 \\ -37 & -16 & 43 \\ 1 & 43 & 11 \end{pmatrix}$$

hat das 3-adische und 5-adische Geschlechtssymbol

$$(1^{-1}9)(1^{-2}25^{-1}).$$

Damit ergibt sich

$$\text{Watson}(A) = \begin{pmatrix} 1 & -1 & 2 \\ -1 & 3 & -2 \\ 2 & -2 & 2 \end{pmatrix}.$$

Modulo $\text{Aut}_{\mathbb{Z}}(\text{Watson}(A))$ existiert nur ein einziger D -perfekter Punkt zu $\text{Watson}(A)$, so dass die Berechnung der Automorphismengruppe von A mit der Verwendung des WATSON-Prozesses (vgl. Abschnitt 5.6) etwa 0,5 Sekunden dauert, während sie ohne dieses Verfahren etwa 200 Sekunden dauerten. Dies ist auch dadurch zu erklären, dass es zu A insgesamt 16 Äquivalenzklassen D -perfekter Punkte gibt.

Erzeuger von $\text{Aut}_{\mathbb{Z}}(A)$ sind gegeben durch

$$\text{Aut}_{\mathbb{Z}}(A) = \left\langle \begin{pmatrix} -1 & 2 & 0 \\ 0 & 1 & 0 \\ 0 & -8 & -1 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 \\ 5 & 1 & -5 \\ 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} -36 & -20 & 11 \\ 35 & 19 & -11 \\ -213 & -114 & 68 \end{pmatrix}, \right.$$

$$\begin{pmatrix} -485 & -330 & 114 \\ 392 & 267 & -92 \\ -2020 & -1378 & 473 \end{pmatrix}, \begin{pmatrix} -255 & -408 & -16 \\ 192 & 307 & 12 \\ -832 & -1326 & -51 \end{pmatrix},$$

$$\left. \begin{pmatrix} -445 & -370 & 74 \\ 360 & 299 & -60 \\ -876 & -730 & 145 \end{pmatrix}, \begin{pmatrix} -2427 & -1568 & 614 \\ 2072 & 1339 & -524 \\ -11420 & -7382 & 2887 \end{pmatrix}, \right.$$

$$\begin{aligned}
 & \left(\begin{array}{ccc} 17 & 4 & -22 \\ 192 & 49 & -264 \\ 48 & 12 & -65 \end{array} \right), \left(\begin{array}{ccc} -113 & -152 & 0 \\ 84 & 113 & 0 \\ -318 & -424 & 1 \end{array} \right) \\
 & \left(\begin{array}{ccc} -9860 & -21318 & -1767 \\ 8131 & 17579 & 1457 \\ -43077 & -93126 & -7718 \end{array} \right), \left(\begin{array}{ccc} 31 & 10 & -50 \\ 144 & 49 & -240 \\ 48 & 16 & -79 \end{array} \right), \\
 & \left(\begin{array}{ccc} 126 & 126 & -13 \\ 333 & 337 & -33 \\ 139 & 140 & -14 \end{array} \right), \left(\begin{array}{ccc} 998 & 794 & -181 \\ 3579 & 2843 & -651 \\ 1321 & 1050 & -240 \end{array} \right) \\
 & \left(\begin{array}{ccc} -493 & -884 & -52 \\ 380 & 681 & 40 \\ -1786 & -3196 & -187 \end{array} \right), \left(\begin{array}{ccc} -3645 & -2870 & 674 \\ 2856 & 2249 & -528 \\ -7724 & -6080 & 1429 \end{array} \right), \\
 & \left(\begin{array}{ccc} -1599 & -1500 & 200 \\ 1248 & 1171 & -156 \\ -3424 & -3210 & 429 \end{array} \right), \left(\begin{array}{ccc} -6271 & -4900 & 1176 \\ 5184 & 4051 & -972 \\ -11840 & -9250 & 2221 \end{array} \right), \\
 & \left(\begin{array}{ccc} -656009 & -1432190 & -120170 \\ 543456 & 1186465 & 99552 \\ -2895768 & -6321992 & -530455 \end{array} \right), \\
 & \left(\begin{array}{ccc} -44475 & -19428 & 86204 \\ 36872 & 16107 & -71468 \\ -196652 & -85906 & 381169 \end{array} \right), \\
 & \left(\begin{array}{ccc} 37595 & 82222 & 6914 \\ 122248 & 267371 & 22484 \\ 46940 & 102662 & 8633 \end{array} \right), \left(\begin{array}{ccc} -659383 & -1436116 & -120142 \\ 545632 & 1188369 & 99416 \\ -2903272 & -6323228 & -528985 \end{array} \right) \\
 & \left(\begin{array}{ccc} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{array} \right) \}.
 \end{aligned}$$

(ii) Sei

$$A = \begin{pmatrix} 10 & 4 & 10 \\ 4 & 6 & 6 \\ 10 & 6 & -3 \end{pmatrix}, \quad A^{-1} = \frac{1}{306} \begin{pmatrix} 27 & -36 & 18 \\ -36 & 65 & 10 \\ 18 & 10 & -22 \end{pmatrix}.$$

Das 3- und 17-adische Geschlechtssymbol von A ist dann

$$(1^{-2}9^{-1})(1^{-2}17^{-1}).$$

Der WATSON-Prozess liefert

$$\text{Watson}(A) = \begin{pmatrix} 1 & -1 & 2 \\ -1 & 3 & -2 \\ 2 & -2 & 2 \end{pmatrix}.$$

Zu $\text{Watson}(A)$ gibt es 4 Äquivalenzklassen D -perfekter Punkte und die Berechnung von $\text{Aut}_{\mathbb{Z}}(A)$ über den WATSON-Prozess dauerte weniger als eine Sekunde, zu A gibt es 16 solche Äquivalenzklassen und die direkte Berechnung von $\text{Aut}_{\mathbb{Z}}(A)$ ohne den WATSON-Prozess dauerte etwa 54 Sekunden.

Die Automorphismengruppe von A ist gegeben durch

$$\begin{aligned} \text{Aut}_{\mathbb{Z}}(A) = \langle & \begin{pmatrix} -23 & 44 & -44 \\ -4 & 7 & -8 \\ 8 & -16 & 15 \end{pmatrix}, \begin{pmatrix} 26 & -33 & -30 \\ 45 & -56 & -50 \\ -27 & 33 & 29 \end{pmatrix}, \begin{pmatrix} -1 & -12 & -60 \\ 0 & -9 & -40 \\ 0 & 2 & 9 \end{pmatrix} \\ & \begin{pmatrix} -10 & 27 & -18 \\ 1 & -4 & 2 \\ 7 & -21 & 13 \end{pmatrix}, \begin{pmatrix} 2 & -5 & -10 \\ 9 & -32 & -86 \\ -9 & 35 & 99 \end{pmatrix}, \begin{pmatrix} -31 & 128 & -68 \\ -72 & 303 & -160 \\ 54 & -230 & 121 \end{pmatrix} \\ & \begin{pmatrix} -70 & -143 & -1114 \\ 45 & 96 & 734 \\ 117 & 247 & 1897 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 0 & -9 & 4 \\ 0 & -20 & 9 \end{pmatrix}, \begin{pmatrix} -1385 & -2156 & -19096 \\ -864 & -1343 & -11904 \\ 198 & 308 & 2729 \end{pmatrix} \\ & \begin{pmatrix} -22 & 37 & -46 \\ 9 & -16 & 18 \\ 27 & -47 & 55 \end{pmatrix}, \begin{pmatrix} 190 & 191 & -134 \\ 117 & 116 & -82 \\ -27 & -27 & 19 \end{pmatrix}, \begin{pmatrix} -25 & 12 & 12 \\ 16 & -9 & -8 \\ -68 & 34 & 33 \end{pmatrix} \\ & \begin{pmatrix} -10 & 17 & 22 \\ -9 & 16 & 22 \\ -27 & 47 & 63 \end{pmatrix}, \begin{pmatrix} -170 & -111 & 102 \\ 215 & 144 & -130 \\ -583 & -385 & 351 \end{pmatrix}, \begin{pmatrix} -55 & 128 & 244 \\ 64 & -151 & -292 \\ -182 & 426 & 817 \end{pmatrix} \\ & \begin{pmatrix} -7487 & 4176 & 3744 \\ 6344 & -3537 & -3172 \\ -22048 & 12296 & 11025 \end{pmatrix}, \begin{pmatrix} -6239 & -2900 & 3440 \\ 5580 & 2591 & -3076 \\ -18702 & -8690 & 10311 \end{pmatrix}, \\ & \begin{pmatrix} -1295 & 2796 & 4908 \\ -756 & 1633 & 2868 \\ -2988 & 6452 & 11327 \end{pmatrix}, \begin{pmatrix} -3185 & 7124 & 13012 \\ -1792 & 4009 & 7324 \\ -7288 & 16302 & 29777 \end{pmatrix} \\ & \begin{pmatrix} -2054 & 5925 & 13770 \\ -1143 & 3296 & 7658 \\ -4689 & 13525 & 31431 \end{pmatrix}, \begin{pmatrix} 7615 & -16864 & -30464 \\ 6524 & -14447 & -26096 \\ -1708 & 3782 & 6831 \end{pmatrix}, \\ & \begin{pmatrix} 1490 & -4047 & -8946 \\ 1183 & -3212 & -7098 \\ -287 & 779 & 1721 \end{pmatrix}, \begin{pmatrix} 439 & -100 & -200 \\ 360 & -81 & -164 \\ -90 & 20 & 41 \end{pmatrix}, \\ & \begin{pmatrix} -130 & 257 & 406 \\ -59 & 116 & 182 \\ -287 & 567 & 895 \end{pmatrix}, \begin{pmatrix} -95 & -24 & 48 \\ -40 & -9 & 20 \\ -208 & -52 & 105 \end{pmatrix}, \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \rangle. \end{aligned}$$

(iii) Betrachte

$$A := \begin{pmatrix} 0 & -7 & -4 \\ -7 & 1 & 4 \\ -4 & 4 & 7 \end{pmatrix}, \quad A^{-1} = \frac{1}{135} \begin{pmatrix} 9 & -33 & 24 \\ -33 & 16 & -28 \\ 24 & -28 & 49 \end{pmatrix}$$

Dann ist $\det A = -3^3 \cdot 5$ und das 3-adische und 5-adische Geschlechtsymbol von A gegeben durch

$$(1^{-2}27^{-1})(1^25^{-1}).$$

Der WATSON-Prozess hat also hier einen Effekt. Er liefert dann eine Matrix

$$\text{Watson}(A) = \begin{pmatrix} 0 & -1 & -1 \\ -1 & 5 & -5 \\ -1 & -5 & 0 \end{pmatrix}$$

mit Determinante $-3 \cdot 5$.

$\text{Watson}(A)$ besitzt modulo seiner Automorphismengruppe 5 perfekte Punkte, A selbst besitzt ebenfalls mindestens 5.

Die Berechnung der Automorphismengruppe von A unter Verwendung des WATSON-Prozesses dauert ca. 28 Sekunden, ohne ihn wurde die Rechnung nach ca. 115 Stunden abgebrochen.

Die Automorphismengruppe von A ist gegeben durch:

$$\begin{aligned} \text{Aut}(A) = \langle & \begin{pmatrix} 7 & 0 & 8 \\ 6 & -1 & 6 \\ -6 & 0 & -7 \end{pmatrix}, \begin{pmatrix} 4 & -1 & 3 \\ 6 & -1 & 6 \\ -3 & 1 & -2 \end{pmatrix}, \begin{pmatrix} 1 & 14 & 0 \\ 0 & -1 & 0 \\ 0 & -8 & 1 \end{pmatrix}, \\ & \begin{pmatrix} 4 & 1 & 5 \\ 0 & 1 & 0 \\ -3 & -1 & -4 \end{pmatrix}, \begin{pmatrix} -4 & 51 & -17 \\ -24 & 287 & -100 \\ -3 & 31 & -12 \end{pmatrix}, \begin{pmatrix} 13 & -2 & 4 \\ 588 & -85 & 168 \\ 252 & -36 & 71 \end{pmatrix}, \\ & \begin{pmatrix} 1 & 0 & 0 \\ 168 & -15 & 28 \\ 84 & -8 & 15 \end{pmatrix}, \begin{pmatrix} -1 & 10 & -4 \\ -18 & 205 & -74 \\ -6 & 72 & -25 \end{pmatrix}, \begin{pmatrix} -112 & 2147 & -565 \\ 6 & -113 & 30 \\ 45 & -855 & 226 \end{pmatrix}, \\ & \begin{pmatrix} -140 & 1927 & -611 \\ 6 & -81 & 26 \\ 51 & -697 & 222 \end{pmatrix}, \begin{pmatrix} 19 & 8 & -12 \\ 450 & 179 & -270 \\ 330 & 132 & -199 \end{pmatrix}, \begin{pmatrix} 31 & 64 & -80 \\ 120 & 239 & -300 \\ 108 & 216 & -271 \end{pmatrix}, \\ & \begin{pmatrix} -40 & 1027 & -237 \\ -324 & 8379 & -1928 \\ -267 & 6899 & -1588 \end{pmatrix}, \begin{pmatrix} -17 & 402 & -96 \\ -1296 & 30481 & -7296 \\ -864 & 20328 & -4865 \end{pmatrix}, \\ & \begin{pmatrix} 136 & 95 & -135 \\ 354 & 243 & -346 \\ 333 & 229 & -326 \end{pmatrix}, \begin{pmatrix} 161 & 22 & -36 \\ 930 & 131 & -214 \\ 798 & 112 & -183 \end{pmatrix}, \begin{pmatrix} 184 & 41 & -65 \\ 3378 & 745 & -1182 \\ 2541 & 561 & -890 \end{pmatrix}, \\ & \begin{pmatrix} 29224 & 3353 & -5537 \\ -1698 & -195 & 322 \\ -12291 & -1411 & 2330 \end{pmatrix}, \begin{pmatrix} 24424 & 12673 & -18607 \\ -1362 & -707 & 1038 \\ -10029 & -5205 & 7642 \end{pmatrix}, \\ & \begin{pmatrix} 164 & 99 & -143 \\ 2220 & 1331 & -1924 \\ 1725 & 1035 & -1496 \end{pmatrix}, \begin{pmatrix} 32 & 3 & -5 \\ 7254 & 703 & -1170 \\ 4557 & 441 & -734 \end{pmatrix}, \\ & \begin{pmatrix} 32 & 3 & -5 \\ 8184 & 743 & -1240 \\ 5115 & 465 & -776 \end{pmatrix}, \begin{pmatrix} 1057 & 782 & -1104 \\ 11040 & 8159 & -11520 \\ 8832 & 6528 & -9217 \end{pmatrix}, \\ & \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \rangle \end{aligned}$$

Bemerkung 6.1.3. Es ist möglich, dass die in der Implementierung verwendeten Magma -Intrinsics auch für vermeintlich kleine Beispiele an ihre Grenzen

stoßen. So besitzt für die Matrix

$$A = \begin{pmatrix} -38 & 27 & 25 & -1 \\ 27 & 38 & 26 & 7 \\ 25 & 26 & 47 & 17 \\ -1 & 7 & 17 & 14 \end{pmatrix}$$

der perfekte Punkt

$$x = (1 \ 1 \ -1 \ 0)$$

3175 kürzeste Vektoren. Der `Subsets`-Befehl in `Magma`, der in der Funktion `Directions` (siehe Anhang B) verwendet wird, stößt dort an seine Grenzen.

Anhang A

Grundlegende Definitionen aus der Graphentheorie

[Dic80, Chapter I] folgend werden hier die hier wichtigsten Begriffe aus der Graphentheorie, insbesondere in Zusammenhang mit Gruppenoperationen auf Graphen, zusammengestellt. Da es sich hier im Wesentlichen um eine Aneinanderreihung von Definitionen handelt und auch keine explizite Referenz auf sie erfolgt, werden diese hier nicht nummeriert.

Ein *Graph* X ist formal eine disjunkte Vereinigung $X = V \cup E$ zweier Mengen $V = V(X) \neq \emptyset$ und $E = E(X)$ zusammen mit zwei Abbildungen

$$\iota, \tau : E \rightarrow V.$$

Dabei heißen V bzw. E die *Knoten-* bzw. *Kantenmenge* von X , die Elemente von V bzw. E heißen die *Knoten* bzw. *Kanten* von X . Zu einer Kante $e \in E$ heißt ιe der *Anfangs-* und τe der *Endknoten* von e . Ist V endlich, so nennt man auch X *endlich*. Falls zu jedem $v \in V$ die *Nachbarschaft* von v ,

$$\delta(v) := \{e \in E \mid \iota e = v \text{ oder } \tau e = v\},$$

endlich ist, so heißt X *lokal endlich*.

Ein $e \in E$ mit $\iota e = \tau e$ heißt eine *Schleife*. Man nennt X *einfach*, falls X keine Schleifen enthält.

Für Teilmengen $W \subseteq V$ und $F \subseteq E$ nennt man den Graphen $Y = (W, F)$ einen *Teilgraphen* von X . Der von einer Teilmenge $W \subseteq V$ *induzierte* Teilgraph ist der Graph $X(W) = (W, E(W))$ mit $E(W) = \{e \in E \mid \iota e, \tau e \in W\}$.

Unter einem *Pfad* P in X versteht man eine endliche Abfolge

$$P = v_0, e_1, v_1, \dots, e_n, v_n$$

von Knoten v_0, \dots, v_n mit Kanten $e_i \in E$, wobei hier die Richtung keine Rolle spielt. Hierbei nennt man v_0 den *Anfangsknoten* und v_n den *Endknoten* von P . Als die *Länge* von P definiert man die Anzahl n der Knoten (mit Vielfachheiten). Zwei Knoten v, w von X heißen *verbunden*, falls es einen Pfad P mit Anfangsknoten v und Endknoten w in X gibt. Gibt es einen solchen Pfad mit Länge 1, so heißen v und w *benachbart*. Offenbar definiert Verbundenheit eine

Äquivalenzrelation auf V und damit auf X , indem man die von den Äquivalenzklassen in V induzierten Teilgraphen von X betrachtet. Die Äquivalenzklassen von X nennt man *Zusammenhangskomponenten* (oder auch *Komponenten*) von X . Falls X nur eine Zusammenhangskomponente besitzt, so nennt man X *zusammenhängend*.

Einen Pfad mit gleichem Anfangs- und Endknoten nennt man einen *Kreis*. Enthält X keinen Kreis, so nennt man X einen *Wald*, ein zusammenhängender Wald ist ein *Baum*. Nach dem Lemma von ZORN gibt es eine maximale Teilmenge $E' \subseteq E$, so dass der Graph $T = (V, E')$ ein Wald ist. Wegen der Maximalität von E' müssen zwei in X verbundene Knoten v, w auch in T verbunden sein. Im Fall, dass X zusammenhängend ist, ist T also ein Baum, der so genannte *Spannbaum* oder *maximale Teilbaum* von X .

Sei nun G eine Gruppe, die auf der Knotenmenge V eines Graphen X operiert. Dann *operiert* G auf X , falls für jedes $e \in E$ und jedes $g \in G$ auch $eg := (\iota_e.g, \tau_e g) \in E$ gilt. In diesem Fall ist der *Stabilisator* einer Kante $e = (v, w) \in E$ genau der Schnitt der Stabilisatoren ihrer Anfangs- und Endknoten,

$$\text{Stab}_G(e) := \text{Stab}_G(v) \cap \text{Stab}_G(w),$$

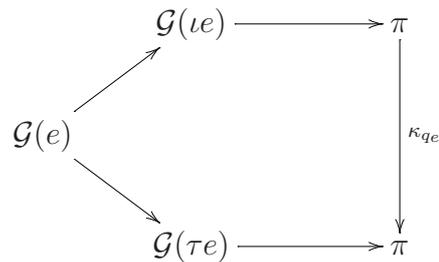
vergleiche hierzu [Ser80, Theorem 6 in §I.4].

Falls der *Restklassengraph* $X/G = (V/G, E/G)$ zusammenhängend ist, lässt er sich zu einer zusammenhängenden Transversale S in X selbst liften (vergleiche hierzu [Dic80, Proposition 2.3]). Für jedes $e = (v, w) \in E(S)$ gibt es dann einen eindeutig bestimmten Knoten $s \in S$, der in derselben G -Bahn liegt wie v , d.h. es gibt für jedes $e \in E(S)$ ein (nicht unbedingt eindeutiges) $q_e \in G$, so dass $vq_e^{-1} = s \in S$. Falls $v \in S$, so wählt man $q_e = 1$. Eine so gebildete Familie $(q_e \mid e \in E(S))$ nennt man eine *verbindende Familie* für S .

Ein *Gruppengraph* $\mathcal{G} : Y \rightarrow \text{Grp}$, abgekürzt auch (\mathcal{G}, Y) , besteht aus folgenden Daten: einem Graphen Y , so dass es zu jedem Knoten $v \in V(Y)$ eine assoziierte Gruppe $\mathcal{G}(v)$ gibt und zu jeder Kante $e \in E(Y)$ jeweils eine Gruppe $\mathcal{G}(e)$, sowie zwei Gruppenhomomorphismen $\iota_e : \mathcal{G}(e) \rightarrow \mathcal{G}(\iota_e)$, $\tau_e : \mathcal{G}(e) \rightarrow \mathcal{G}(\tau_e)$. Die Gruppen $\mathcal{G}(v)$ bzw. $\mathcal{G}(e)$ für $v \in V(Y)$, $e \in E(Y)$ nennt man *Knoten-* bzw. *Kantengruppen* von \mathcal{G} . Falls Y zusammenhängend ist, nennt man auch \mathcal{G} *zusammenhängend*.

Ist (\mathcal{G}, Y) ein zusammenhängender Gruppengraph und T ein Spannbaum für Y . Dann ist die *Fundamentalgruppe* $\pi = \pi(\mathcal{G}, T)$ von \mathcal{G} bezüglich T definiert über folgende universelle Eigenschaft: Für jedes $v \in V(Y)$ gibt es einen Gruppenhomomorphismus $\mathcal{G}(v) \rightarrow \pi$ und zu jedem $e \in E(Y)$ gibt es ein $q_e \in \pi$, so dass der innere Automorphismus $\kappa_{q_e} : \pi \rightarrow \pi, p \mapsto q_e^{-1}pq_e$ das folgende

Diagramm¹ kommutativ macht und $q_e = 1$, falls $e \in E(T)$.



Die Fundamentalgruppe π ist damit präsentiert durch

- Erzeuger:

- die Elemente von $\mathcal{G}(v)$ für $v \in V(Y)$;
- die Elemente q_e , $e \in E(Y)$;

- Relationen:

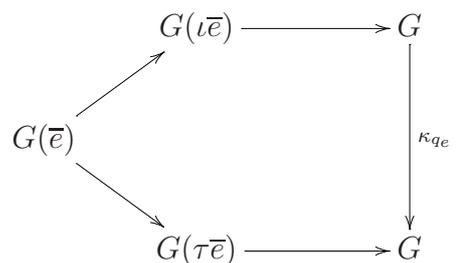
- Relationen von $\mathcal{G}(v)$, $v \in V(Y)$;
- Relationen entsprechend zu

$$(g\iota_e)^{q_e} = g\tau_e \text{ für alle } g \in \mathcal{G}(e), e \in E(Y);$$

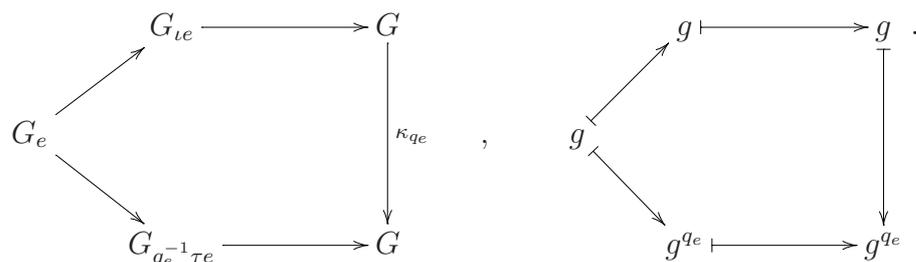
- Relationen entsprechend zu $q_e = 1$ für $e \in E(T)$;

Wie der Satz von BASS und SERRE (vgl. Satz 3.2.9) aussagt, ist das im Wesentlichen kanonische Beispiel für einen Gruppengraphen das folgende:

Es sei G eine Gruppe, die auf dem Graphen X operiere, so dass X/G zusammenhängend ist, S eine zusammenhängende Transversale und $(q_e \mid e \in E(S))$ eine verbindende Familie. Mit jedem Knoten \bar{s} von X/G ($s \in S$) kann man nun die Gruppe $G(\bar{s}) := G_s$ assoziieren. Dann ergibt sich zu jeder Kante \bar{e} von X/G ($e \in E(S)$) das folgende kommutative Diagramm



genauer



¹die Diagramme in diesem Kapitel wurden mit dem L^AT_EX-Paket xy-pic erzeugt

Anhang B

Magma-Code

B.1 Der Automorphismen-Algorithmus

```
1 freeze;

/******
 * Calculating the automorphism group of a hyperbolic lattice *
6 * See *
 * J. Opgenorth, "Dual Cones and the Voronoi Algorithm", *
 * [Opg] *
 * M.H. Mertens, "Duale Kegel und Automorphismengruppen *
 * hyperbolischer Gitter", [Mert] *
11 * *
 * Michael H. Mertens, September 2012 *
*****/

16 PerfectPointsFormat:=recformat<
    ConeTestVectors,
    Points,
    Lat,
21 directions,
    Neighbours,
    Stabilizers,
    Connectors,
    AutGrp,
26 Complete
    >;

ZZ:=Integers();
31 QQ:=Rationals();
// RR:=RealField(20);

// First we have to be able to determine whether a given matrix
36 // is hyperbolic or not

function SylvesterType(A)
// The Sylvester type of A is given as a tuple (p,-q),
```

```

41 // where p denotes the number of positive, q the number of
// negative eigenvalues of A.

    if not (IsSymmetric(A) and (Type(BaseRing(A)) eq RngInt or
        Type(BaseRing(A)) eq FldRat)) then
46     error "A must be a real symmetric matrix";
    end if;
    D,_,_:=OrthogonalizeGram(A);
    Spec:=[D[i,i] : i in [1..Nrows(A)]];
    p:=#[ s : s in Spec | s gt 0];
51 q:=#[ t : t in Spec | t lt 0];
    return <p,-q>;
end function;

56 function IsHyperbolic( A )
// Returns true, if a symmetric matrix is of Sylvester type
// (n-1,-1), false otherwise
    t:=SylvesterType(A);
    n:=NumberOfRows(A);
61 return t eq <n-1,-1>;
end function;

function IsIntegralMatrix( A )
66 if Type(BaseRing(A)) eq RngInt then
    return true;
    elif Type(BaseRing(A)) eq FldRat then
        ADen:=[Denominator(a) : a in Eltseq(A)];
        l:=LCM(ADen);
71 return l eq 1;
    else
        error "Can not determine whether argument is integral or not";
    end if;
end function;

76

is_even := function(A)
    return GCD(Diagonal(A)) mod 2 eq 0;
81 end function;

function split_odd(A) //Nach Kneser, Satz 26.3 Seite 107
    n0:=NumberOfRows(A);
86 if n0 eq 1 then
    return A[1,1], 0;
    end if;
    even:=true;
    for j in [1..n0] do
91 if GF(2) ! A[j][j] ne 0 then
        i:=j; even:=false;
        break;
        end if;
    end for;
96
    if even then print "Fehler"; end if;
    C := RMatrixSpace(Integers(8),n0-1,n0) ! 0;

```

```

//u := Integers() ! (Integers(32) ! A[i,i]^-1);
u:=A[i][i]; //u^-1 mod 8
101 for j in [1..n0-1] do
    if j lt i then
        C[j,j] := 1;
        C[j,i] := -A[j,i]*u ;
106     else
        C[j,j+1] := 1;
        C[j,i] := -A[j+1,i]*u ;
        end if;
    end for;

111 B := C*A*Transpose(C);
    even:=true;
    for j in [1..n0-1] do
        if GF(2) ! B[j][j] ne 0 then
116         i1:=j; even:=false;
            break;
        end if;
    end for;

121 if even then
    for j in [1..n0-1] do
        for k in [j+1..n0-1] do
            if GF(2) ! B[j][k] ne 0 then
126             j0:=j; k0:=k;
                break j;
            end if;
        end for;
    end for;
    I:=MatrixRing(Integers(8),n0) ! 0;

131 for j in [1..n0] do
    I[1][j] := C[j0][j];
end for;

136 I[1][i] += 1;

    for k in [2..n0] do
        for j in [1..n0] do
            I[k][j]:=C[k-1][j];
141        end for;
    end for;

    A := I*A*Transpose(I);
    C := RMatrixSpace(Integers(8),n0-1,n0) ! 0;
146 u:=A[1][1]; //u^-1 mod 8

    for j in [1..n0-1] do
        C[j,j+1] := 1;
        C[j,1] := -A[j+1,1]*u ;
151    end for;

    B := C*A*Transpose(C);
    end if;
    return u, B;
156 end function;

```

```

function twoDiagonalization(A)
  n:=NumberOfRows(A);
161  A8:=MatrixRing(Integers(8),n) ! A;
  AD:=Parent(A) ! 0;
  B:=A8;

  for i in [1..n] do
166    u,B := split_odd(B);
    AD[i][i]:=Integers() ! u;
  end for;

  return AD;
171 end function;

function two_adic_genus_symbol(A, bound)
  error if not IsSymmetric(A), "A is not a symmetric matrix.";
176  p:=2;
  n := NumberOfRows(A);
  F_n := MatrixAlgebra(GF(p), n); // F_p^{n \times n}
  V_n := VectorSpace(GF(p), n); // F_p^n
  Z_n := MatrixAlgebra(Integers(), n); // Z^{n \times n}
181  Z8_n := MatrixAlgebra(Integers(8), n); // (Z/8Z)^{n \times n}
  A := ChangeRing(A, Integers()); // allows A to be formally
  // defined over Rationals()
  // upon calling this routine
  // Divide A by the highest power of p dividing all its elements:
186  g := Content(A); // gcd of all elements of A
  v := Valuation(g, p);
  A := A div (p^v);
  A_p := F_n!A; // A over F_2
191  C := Kernel(A_p);

  // Otherwise, we split off the Jordan constituent of the highest
  // power of 2:
  if Dimension(C) eq 0 then
196    e0 := KroneckerSymbol(Determinant(A), 2);
    if is_even(A) then
      s := 0; // "even"
      o := 0; // oddity not applicable for even lattices,
      // stored as 0
201    else
      s := 1; // "odd"
      // An odd matrix can be diagonalized over the 2-adic integers.
      // We can then determine the oddity by computing the trace
      // of Z/8Z:
206    AD := twoDiagonalization(A);
      o := Integers(8)!Trace(AD);
    end if;
    return [[v, n, e0, s, Integers()!o]];
  else
211  T := ChangeRing(BasisMatrix(C), Integers());
  B_p := BasisMatrix(Complement(V_n, C));
  B := ChangeRing(B_p, Integers());
  AB := B*A*Transpose(B);

```

```

ABI:=Adjoint(AB);
216 det:=Determinant(AB);
deti:=Integers() ! ((Integers(p^bound) ! det) ^-1);
X:=B*A*Transpose(T);
KOR:= deti*Transpose(X)*ABI*X;
J2 := T * A * Transpose(T)-KOR;
221 n0 := Rank(B_p);

if is_even(AB) then
    s:=0;
    o:=0;
226 else
    s:=1;
    //to compute the oddity diagonalize the matrix over the
    // p-adic integers and then compute the trace over
    // Z mod 8
231 AD:=twoDiagonalization(AB);
    o:=Integers(8) ! Trace(AD);
end if;

return [[v,n0,KroneckerSymbol(Determinant(AB),p),s,Integers(!o)]
236 cat $$ (p^v*J2,bound);
end if;
end function;

241 function p_adic_genus_symbol(A, p,bound)
error if not IsSymmetric(A), "A is not a symmetric matrix";
error if not IsPrime(p), "p is not prime";

// parameter "bound": should be >= 3 + p-valuation of the maximal
246 // elementary divisor of (passed as a parameter since p_adic_genus
// is usually called multiple times for the same matrix and
// calculating elementary divisors is expensive)

if p eq 2 then
251 return two_adic_genus_symbol(A, bound);
end if;

n := NumberOfRows(A);
F_n := MatrixAlgebra(GF(p), n); // F_p^{n \times n}
256 V_n := VectorSpace(GF(p), n); // F_p^n
Z_n := MatrixAlgebra(Integers(), n); // Z^{n \times n}
A := ChangeRing(A, Integers()); // allows A to be formally defined
// over Rationals() upon calling
// this routine

261 v := Valuation(Content(A), p);
A := A div (p^v);
A_p := F_n!A; // A over F_p
C := Kernel(A_p);
// If the kernel of A mod p is trivial, A consists only of a
266 // zero-valuationed Jordan constituent. Otherwise, we split off
// the Jordan constituent of the highest p-power:

if Dimension(C) eq 0 then
    e0 := KroneckerSymbol(Determinant(A), p);
271 // printf "Only one Jordan constituent,
// stopping iteration. p^v=%o.\n", p^v;

```

```

    return [[v, n, e0]];
else
  T := ChangeRing(BasisMatrix(C), Integers());
276 B_p := BasisMatrix(Complement(V_n, C));
  A_p := B_p * A_p * Transpose(B_p);
  B := ChangeRing(B_p, Integers());
  AB := B*A*Transpose(B);
  ABI:=Adjoint(AB);
281 det:=Determinant(AB);
  deti:=Integers() ! ((Integers(p^bound) ! det) ^-1);
  X:=B*A*Transpose(T);
  KOR:= deti*Transpose(X)*ABI*X;
  J2 := T * A * Transpose(T)-KOR;
286 n0 := Rank(A_p);
  if IsSquare(Determinant(A_p)) then
    e0 := 1;
  else
    e0 := -1;
291 end if;
  // printf "Iterating with p^v=%o, matrix A =\n%o\n,
  // matrix A_p =\n%o\nmatrix J2 = \n%o\nbasis of
  // kernel = \n%o\nbasis of
  // complement = \n%o\n\n",p^v,A,A_p,J2,BasisMatrix(C),
296
  return [[v, n0, e0]] cat $$ (p^v * J2, p, bound);
end if;
end function;

301 function genus_symbol(A)
  if not(A in RMatrixSpace(IntegerRing(), NumberOfRows(A),
    NumberOfColumns(A))) or
    (NumberOfRows(A) ne NumberOfColumns(A)) or
306 not(IsSymmetric(A)) then
    error "Argument gram_matrix must be a symmetric positive definite
    integral matrix.";
  end if;

311 divisors := [d[1]: d in Factorization(Determinant(A)) | d[1] gt 2];
  g := ElementaryDivisors(A);
  h := Factorization(g[#g]);
  sym := [<2, p_adic_genus_symbol(A, 2, Valuation(g[#g], 2)+3)>];

316 for t in h do
  if t[1] ne 2 then
    Append(~sym, <t[1], p_adic_genus_symbol(A, t[1], t[2])>);
  end if;
end for;

321 return sym;
end function;

326 // Watson process

function WatsonGram(A, p)
  // Valuation(Determinant(A), p);

```

```

331  D:= p * ChangeRing(A, QQ)^-1;
      L:= Lattice(D);
      P:= (L meet StandardLattice(Ncols(A)));
      T:= Matrix(QQ, BasisMatrix(P));
      AA:= p^-1*T * A * Transpose(T);
336  // Valuation(Determinant(AA), p);
      return ChangeRing(AA,ZZ), T;
      end function;

341  function WatsonProcess(A)
      n:=Nrows(A);
      B:=A;
      C:=IdentityMatrix(QQ,n);
      sym:=genus_symbol(A);
346  for p in sym do
      q:=p[1];
      a:=Max([p[2][i][1] : i in [1..#p[2]]]);
      while a ge 2 do
351      B,T:=WatsonGram(B,q);
          C:=T*C;
          a-:=2;
      end while;
      if Valuation(Determinant(B),q) gt
356      Valuation(Determinant(A),q) then
          B,T:=WatsonGram(B,q);
          C:=T*C;
      end if;
      end for;
361  return B, C;
      end function;

366  function Get_y( A : record:=rec< PerfectPointsFormat | > )
      // finds a vector y s.t. yA^-1y^tr<0 for a rational matrix A
      if not assigned record'ConeTestVectors then
      A:=ChangeRing(A,QQ);
      ok, Ainv:=IsInvertible(A);
371  if not ok then
          error "Error: A must be invertible.";
      end if;
      D,T,_:=OrthogonalizeGram(A);
      SpecNeg:=[ i : i in [1..Nrows(A)] | D[i,i] lt 0];
376  if SpecNeg eq [] then
          error "Error: There is no such y.";
      end if;
      y1:=Rows(Transpose(T)^-1)[SpecNeg[1]];
      y2:=Rows(T)[SpecNeg[1]];
381  return [y1,y2];
      else return record'ConeTestVectors;
      end if;
      end function;

386  function xAextr(x,A)
      x:=ChangeRing(x,QQ);

```

```

    A:=ChangeRing(A,QQ);
    return (Vector(x) * A * Matrix(1, Eltseq(x)))[1];
391 end function;

function GetConeVector( A : record:=rec< PerfectPointsFormat | > )
396 if not (IsHyperbolic(A) and IsIntegralMatrix(A)) then
    error "Argument must be an integral hyperbolic matrix";
end if;
record'ConeTestVectors:=Get_y(A : record:=record);
y1:=record'ConeTestVectors[1];
401
n:=Nrows(A);
ZZn:=StandardLattice(n);
Bound:=0;
x:=Vector([0^n]);
406 null:=x;

while x eq null do
    Bound+=1;
    S:=[Vector(s[1]) : s in ShortVectors(ZZn, Bound)];
411 for y in S do
    if xAxtr(y,A) lt 0 and xAxtr(y,A) ge -100 then
        x:=y;
    end if;
end for;
416 end while;

if (ChangeRing(x,QQ),ChangeRing(y1,QQ)) lt 0 then
    x:=-x;
end if;
421 return ChangeRing(x, ZZ);
end function;

function xperp(x)
426 if Type(BaseRing(x)) eq FldRat then
    xDen:=[Denominator(xi) : xi in Eltseq(x)];
    r:=LCM(xDen);
    x:=ChangeRing(r*x, ZZ);
elif Type(BaseRing(x)) eq RngInt then
431 x:=Vector(RemoveRowContents(Matrix(x)));
else
    error "Argument must be a rational vector.";
end if;
H:= Basis(NullSpace(Matrix(ZZ, 1,Eltseq(x))));
436 assert Max(ElementaryDivisors(Matrix(H))) eq 1;
return H;
end function;

441 // This is an exact copy of the corresponding function in the
// normalizer package:
// G acts through h: SxG->S. We find generators for the stabilizer
// in G of some X \in S. G is usually infinite!
// But the orbit of X is not.
446

```

```

function MyStabGens(G, h, X)
  n:= Ngens(G);
  Orbit:= {@ X @};
  Path:= [ Id(G) ];
451  Stab:= [];

  i:= 0;
  while i lt #Orbit do
    i+= 1;
456  for j in [1..n] do
    x:= Path[i] * G.j;
    Xx:= h(X, x);
    idx:= Index(Orbit, Xx);
    if idx eq 0 then
461      Include(~Orbit, Xx);
      Append(~Path, x);
    else
      x:= x * Path[idx]^-1;
      assert h(X, x) eq X;
466      if not IsOne(x) and x notin Stab and x^-1 notin Stab then
        Append(~Stab, x);
      end if;
    end if;
  end for;
471  end while;

  return Stab;
end function;

476

// now we start:

/*****
481 * For given integer x in the cone  $V_1^{\{>0\}}$  one can calculate the set *
* of D-minimal vectors for x by the following method:
*
*  $A^{-1}$  defines a positive definite scalar product on the hyperplane *
* H perpendicular (w.r.t the standard scalar product) to x. Define
*
* the lattice  $L= H \cap 1/(-N(x))\mathbb{Z}^n$  endowed with the scalar product
*
486 * induced by  $A^{-1}$ . Here  $N(x)=xAx^{\text{tr}}$ . For  $\lambda=2/(-N(x)), 3/(-N(x)), \dots$ 
*
* calculat the set  $M_\lambda$  of lattice points y with
*
*  $yA^{-1}y^{\text{tr}} \leq -\lambda^2 N(x)$ 
*
* As soon as this set is not empty (which is the case at least for
*
*  $\lambda=1$ ), the set  $\{-\lambda xA + y \mid y \text{ in } S_L\} \cap \mathbb{Z}^n$  is the
*
491 * desired answer.
*
*****/

function DMinimalVectors(x, A : record:=rec< PerfectPointsFormat | > )
  record 'ConeTestVectors:=Get_y(A : record:=record);

```

```

496 y1:=record'ConeTestVectors[1];
y2:=record'ConeTestVectors[2];
if not (Type(BaseRing(x)) eq RngInt or
        Type(BaseRing(x)) eq FldRat) then
    error "First argument must be a rational vector";
501 elif not (IsIntegralMatrix(A) and IsHyperbolic(A)) then
    error "Second argument must be an integral hyperbolic matrix.";
elif not (xAxtr(x,A) lt 0 and
          (ChangeRing(x,QQ),ChangeRing(y1,QQ)) gt 0) then
    error "First argument does not belong to the dual cone.";
506 end if;
// First rescale x s.t. it becomes an integer vector
// Note that the D-minimum of a vector x is rescaled by lambda
// and the set of D-minimal vectors remains the same if one
// replaces x by lambda*x, lambda > 0
511 if Type(BaseRing(x)) eq FldRat then
    xSeq:=Eltseq(x);
    xDen:=[Denominator(xi) : xi in xSeq];
    r:=Lcm(xDen);
516 else
    r:=1;
end if;
x:=ChangeRing(r*x, ZZ);
y2:=ChangeRing(y2,QQ);
521 H:=xperp(x);
x:=ChangeRing(x, QQ);
A:=ChangeRing(A, QQ);
Ainv:=A^-1;
N:=-xAxtr(x,A);
526 B:=1/N*ChangeRing(Matrix(H),QQ);
L:=LatticeWithGram(B*Ainv*Transpose(B));
ZZn:=StandardLattice(Nrows(A));

if N ge 3000 or Determinant(GramMatrix(L)) ge 10000 then
531 N, Determinant(GramMatrix(L));
    print "Warning: This might take a while.";
end if;

// Of course Ainv is not positive definite on RR^n,
536 // but it is indeed on H.
l:=0;
M:=[];
while M eq [] do
    l+=1;
541 lambda:=1/N;
    S:=ShortVectors(L, lambda^2*N);
    S1:=[Vector(ChangeRing(s[1],QQ)*B) : s in S];
    S2:=[-s : s in S1];
    S:=S1 cat S2 cat [Vector(ChangeRing(L!0,QQ)*B)];
546
    for i in [1..#S] do
        y:=S[i];
        d:=y-lambda*x*A;
        if d in ZZn and (y2,ChangeRing(d,QQ)) ge 0 then
551 Append(~M, d);
        end if;
    end for;
end while;

```

```

end while;

556 mu:=(ChangeRing(x,QQ),ChangeRing(M[1],QQ));
    return M, mu/r;
end function;

561
// This is a rather primitive method to find D-minimal vectors,
// should only be used for debugging

function DMinimalVectors2(x, A, Bound :
566     record:=rec<PerfectPointsFormat | >)
    n:= Nrows(A);
    x:= ChangeRing(x, QQ);
    C:= CartesianProduct([ [ -Bound..Bound ]^^n ]);
    mu:= Infinity();
571 M:= [];
    record'ConeTestVectors:= Get_y(A : record:=record);
    y:=record'ConeTestVectors;
    for c in C do
        v:= Vector(QQ, [x: x in c ]);
576     if xAextr(v, ChangeRing(A,QQ)^-1) le 0 and
        (Vector(QQ,y[2]),v) ge 0 and v ne Parent(v)!0 then
            t:= (x,v);
            if t lt mu then mu:= t; M:= [v];
            elif t eq mu then Append(~M, v);
581         end if;
        end if;
    end for;
    return M, mu;
end function;

586

// We do almost the same to calculate the set of D-vectors with
// (x, d) le C, C > 0.
591 function DShortVectors(x, A, C :
        record:=rec<PerfectPointsFormat | >)
    if not C gt 0 then
        error "Bound must be positive.";
    end if;

596 record'ConeTestVectors:=Get_y(A : record:=record);
y1:=record'ConeTestVectors[1];
y2:=record'ConeTestVectors[2];

601 if not (Type(BaseRing(x)) eq RngInt or
        Type(BaseRing(x)) eq FldRat) then
    error "First argument must be a rational vector";
elif not (Type(BaseRing(A)) eq RngInt and IsHyperbolic(A)) then
    error "Second argument must be an integral hyperbolic matrix.";
606 elif not (xAextr(x,A) lt 0 and
        (ChangeRing(x,QQ),ChangeRing(y1,QQ)) gt 0) then
    error "First argument does not belong to the dual cone.";
end if;
// First rescale x s.t. it becomes an integer vector

611

```

```

if Type(BaseRing(x)) eq FldRat then
  xSeq:=Eltseq(x);
  xDen:=[Denominator(xi) : xi in xSeq];
  r:=Lcm(xDen);
616 else
  r:=1;
  end if;
  x:=ChangeRing(r*x, ZZ);
  H:=xperp(x);
621 x:=ChangeRing(x, QQ);
  A:=ChangeRing(A, QQ);
  Ainv:=A^-1;
  N:=-xAxtr(x,A);
  B:=1/N*ChangeRing(Matrix(H),QQ);
626 L:=LatticeWithGram(B*Ainv*Transpose(B));
  ZZn:=StandardLattice(Nrows(A));
  // Of course Ainv is not positive definite on RR^n,
  // but it is indeed on H.
  l:=0;
631 M:=[];
  mu:=0;
  while mu lt r*C do
    l+=1;
    lambda:=l/N;
636 S:=ShortVectors(L, lambda^2*N);
    S1:=[Vector(ChangeRing(s[1],QQ)*B) : s in S];
    S2:=[-s : s in S1];
    S:=S1 cat S2 cat [Vector(ChangeRing(L!0,QQ)*B)];
641 for i in [1..#S] do
    y:=S[i];
    d:=y-lambda*x*A;
    if d in ZZn and (ChangeRing(y2,QQ),d) ge 0 then
      Append(~M, d);
646 mu2:=(ChangeRing(x,QQ),ChangeRing(d,QQ));
      if mu2 gt mu then
        mu:=mu2;
      end if;
    end if;
651 end for;
  end while;
  return M;
end function;

656

function IsDPerfect( x, A : M:=[],
  record:=rec<PerfectPointsFormat | > )
  if M eq [] then
661 M,_:=DMinimalVectors(x, A : record:=record);
  end if;
  V:=VectorSpace(QQ, Ncols(x));
  U:=sub< V | M >;
  return Dimension(U) eq Dimension(V);
666 end function;

```

```

function Directions( x, A : M:=[],
671     record:=rec<PerfectPointsFormat | >)
  if not IsDPerfect(x, A : record:=record) then
    error "First argument must be a D-perfect vector.";
  end if;
  n:=Nrows(A);
676  if M eq [] then
    M, mu:=DMinimalVectors(x,A);
  else
    mu:=(ChangeRing(x,QQ),ChangeRing(M[1],QQ));
  end if;
681  M:=SequenceToSet(M);
  S:=Subsets(M, n-1);
  directions:={ };
  for T in S do
    TT:=Matrix(SetToSequence(T));
686    Tperp:=Basis(Nullspace(Transpose(TT)));
    if #Tperp eq 1 then
      r:=Tperp[1];
      Tc:=M diff T;
      if forall{ z : z in Tc | (z, r) le 0} then
691        r:=-r;
      end if;
      if forall{ z : z in Tc | (z, r) ge 0} then
        // r is now a direction of x, but maybe r is blind:
        y1:=Get_y(A : record:=record)[1];
696        if not (xAxtr(r,A) le 0 and
          (ChangeRing(r,QQ),ChangeRing(y1,QQ)) ge 0) then
          // we want the directions to be integral vectors
          if Type(BaseRing(r)) eq FldRat then
            rDen:=[Denominator(ri) : ri in Eltseq(r)];
701            l:=Lcm(rDen);
          else
            l:= 1;
          end if;
          Include(~directions, ChangeRing(l*r, ZZ));
706        else
          print "Found blind direction:";
          r;
          end if;
        end if;
711      end if;
    end for;
    return SetToSequence(directions);
  end function;

716
// This function adapts the function vor_neighbor from the
// normalizer package
function VoronoiNeighbour1(A, x, r : Limit:=10^6,
  record:=rec<PerfectPointsFormat | >)
721  record' ConeTestVectors:=Get_y(A : record:=record);
  Mx,mux:=DMinimalVectors(x,A : record:=record);
  // rescale x s.t. it becomes integral
  if Type(BaseRing(x)) eq FldRat then
    xSeq:=Eltseq(x);
726    xDen:=[Denominator(xi) : xi in xSeq];
    l:=Lcm(xDen);

```

```

else
  l:=1;
end if;
731 if Type(BaseRing(r)) eq FldRat then
  rSeq:=Eltseq(r);
  rDen:=[Denominator(ri) : ri in rSeq];
  lr:=Lcm(rDen);
else
736   lr:=1;
end if;

t1:=Get_y(A : record:=record)[1];
// Just to make sure nothing awkward happens...
741 x:=ChangeRing(x*l, ZZ);
muy:=ZZ!(muy*l);
r:=ChangeRing(lr*r, ZZ);
xr:= ChangeRing(x,QQ);
rr:= ChangeRing(r,QQ);

746 // Calculate a vector y=lm*x+rm*r which lies in the cone and
// whose D-minimum is smaller than muy

lu:= 1; ru:= 0;
751 lo:= 0; ro:= 1;
found:= false;
counter:=0;
repeat
  lm:=lu + lo;
756  rm:=ru + ro;
  y:=lm*xr + rm*rr;
  if xAxtr(y,A) lt 0 and (y,ChangeRing(t1,QQ)) gt 0 then
    My, muy:=DMinimalVectors(y,A);
    if muy lt lm*muy then
761     found:= true;
    else
      lu:= lm; ru:= rm;
    end if;
  else
766   lo:= lm; ro:= rm;
  end if;
  counter +=1;
until found or counter ge Limit;

771 if counter ge Limit then // This should never happen
  error "After",Limit," trials there was no suiting point.";
end if;

/*****
776 * If y1 is a D-minimal vector for y, then by construction *
*   lm*muy > (y,y1) = lm*(x,y1) + rm*(r,y1) *
* Thus since (x,y1) > muy it holds that (r,y1) < 0. *
* We calculate the rational number *
*   p/q= (muy-(x,y1))/(r,y1)>=0. *
781 * Then (q*x+p*r,y1)=q*muy and we replace y by q*x+p*r. *
* If then muy= q*muy we are done, otherwise repeat *
*****/

repeat

```

```

786   y1:=ChangeRing(My[1], QQ);
      xy1:=ZZ!(Vector(x)*Matrix(1,Eltseq(y1)))[1];
      ry1:=-ZZ!(Vector(r)*Matrix(1,Eltseq(y1)))[1];

      xy1 -= ZZ!mux;
791
      d:=GCD(xy1,ry1);
      if d ne 1 then
        xy1 div:= d;
        ry1 div:= d;
796   end if;

      y:=ry1*x+xy1*r;
      My, muy:=DMinimalVectors(y,A : record:=record);

801   until muy eq ry1*mux;

      if IsDPerfect(x,A : M:=Mx, record:=record) then
        assert IsDPerfect(y,A : M:=My, record:=record);
      end if;

806   return y, My,muy;
end function;

811 // This function is supposed to do the same as VoronoiNeighbour1,
    // but uses a slightly different method

function VoronoiNeighbour2(A,x,r : Limit:=10^3,
                          record:=rec<PerfectPointsFormat | >)
816   record'ConeTestVectors:=Get_y(A : record:=record);
      if xAextr(x,A) ge 0 or
        (ChangeRing(x,QQ),record'ConeTestVectors[1]) le 0 then
        error "2nd argument does not belong to the cone.";
      end if;
821 // rescale x s.t. it becomes integral
      if Type(BaseRing(x)) eq FldRat then
        xSeq:=Eltseq(x);
        xDen:=[Denominator(xi) : xi in xSeq];
        l:=Lcm(xDen);
826   else
        l:=1;
      end if;

      if Type(BaseRing(r)) eq FldRat then
831   rSeq:=Eltseq(r);
        rDen:=[Denominator(ri) : ri in rSeq];
        lr:=Lcm(rDen);
      else
        lr:=1;
836   end if;

      t1:=record'ConeTestVectors[1];
      yA:= ChangeRing(record'ConeTestVectors[1], QQ);

841 // Just to make sure nothing awkward happens...
      x:=ChangeRing(l*x, ZZ);
      Mx,mux:=DMinimalVectors(x,A);

```

```

r:=ChangeRing(lr*r, ZZ);
xr:= ChangeRing(x,QQ);
846 rr:= ChangeRing(r,QQ);

// Calculate a vector y=lm*x+rm*r which lies in the cone and
// whose D-minimum is smaller than lm*mux
lu:= 1; ru:= 0;
851 lo:= 0; ro:= 1;
lm:=0;
found:= false;
counter:=0;
repeat
856 counter+=1;
lm:=lu + lo;
rm:=ru + ro;
y:=lm*xr + rm*rr;
if xAxtr(y,A) lt 0 and (y,yA) gt 0 then
861 My, muy:=DMinimalVectors(y,A : record:=record);
if muy lt lm*mux then
found:= true;
else
lu:= lm; ru:= rm;
866 end if;
else
lo:= lm; ro:= rm;
end if;
until found or counter ge Limit;
871 if counter ge Limit then // This should never happen
error "After",Limit," trials there was no suiting point.";
end if;
M:=DShortVectors(y,A,lm*mux : record:=record);
M:= [ x: x in M | x notin Mx ];
876 assert forall{d: d in M | (Vector(r)*Matrix(1,Eltseq(d)))[1] lt 0 };
lambda:=Min([ (mux - ZZ!(Vector(x)*Matrix(1,Eltseq(d)))[1])
/(Vector(r)*Matrix(1,Eltseq(d)))[1] : d in M ]);
p:=Numerator(lambda);
q:=Denominator(lambda);
881 y:=q*x+p*r;
My, muy:=DMinimalVectors(y,A : record:=record);

if IsDPerfect(x,A : M:=Mx, record:=record) then
assert IsDPerfect(y,A : M:=My, record:=record);
886 end if;

return y, My,muy;
end function;

891 // L : Basis of Q^n
function action(L, g)
B:= Matrix([ ChangeRing(l,QQ) * ChangeRing(g, QQ) : l in L ]);
d:= Denominator(B);
896 M:= HermiteForm( Matrix(ZZ, d*B) );
return [ ChangeRing(M[i], QQ) / d : i in [1..Nrows(M)] ];
end function;

901

```

```

function PerfectStab(x,A)
  if Type(BaseRing(x)) eq FldRat then
    xDen:=[Denominator(xi) : xi in Eltseq(x)];
    l:=Lcm(xDen);
906  else
    l:=1;
    end if;

  x:=ChangeRing(l*x,ZZ);
911  n:=Nrows(A);
  ZZn:=Basis(StandardLattice(n));
  ZZn:=[ChangeRing(Vector(l),QQ) : l in ZZn];
  H:=Matrix(xperp(x*A));
  C:=H*A*Transpose(H);
916  L:=LatticeWithGram(C);
  Aut:=GeneratorsSequence(AutomorphismGroup(L));
  Aut:=[DiagonalJoin(g, Matrix(1,[1])) : g in Aut ];
  B:=ChangeRing(VerticalJoin(H, Matrix(x)), QQ);
  GxGens:=[B^-1*g*B : g in Aut ];
921  Gx:=sub< GL(n,QQ) | GxGens>;

  // Gx is not a unimodular group in general, but the stabilizer
  // Sx of a D-perfect point can be found as a subgroup via orbit
  // algorithm. Sometimes, there is a shortcut if the generators
926  // of Gx are all integral.

  if forall{ g : g in GxGens | IsIntegralMatrix(g) } then
    Sx:=sub< GL(n,ZZ) | GxGens >;
    Sxrat:=Gx;
931  else
    SxGens:=MyStabGens(Gx, action, ZZn);
    assert forall{ s : s in SxGens | s*ChangeRing(A,QQ)*Transpose(s)
      eq ChangeRing(A,QQ)};
    Sx:=sub< GL(n,ZZ) | SxGens >;
936  Sxrat:=sub< Gx | SxGens >;
    end if;

  return Sx, Transversal(Gx, Sxrat), LLLGramMatrix(L);
end function;
941

function PerfectConnector(x, y, A : Trans:={@ @})
  if Type(BaseRing(x)) eq FldRat or
    Type(BaseRing(y)) eq FldRat then
946  xDen:=[Denominator(xi) : xi in Eltseq(x)];
    lx:=Lcm(xDen);
    yDen:=[Denominator(yi) : yi in Eltseq(y)];
    ly:=Lcm(yDen);
  else
951  lx:=1;
    ly:=1;
    end if;

  x:=ChangeRing(lx*x,ZZ);
956  y:=ChangeRing(ly*y,ZZ);

  // rescale x and y s.t. xAx^tr=yAy^tr
  xAx:=-ZZ!(xAxtr(x,A));

```

```

    yAy:=-ZZ!(xAxtr(y,A));
961  l:=Gcd(xAx,yAy);
    okx,lx:=IsSquare(yAy div l);
    oky,ly:=IsSquare(xAx div l);
    if not (okx and oky) then
        return false,_;
966  end if;

    x:=(ZZ!lx)*x;
    y:=(ZZ!ly)*y;

971  Hx:=Matrix(xperp(x*A));
    Cx:=Hx*A*Transpose(Hx);
    Lx:=LatticeWithGram(Cx);

    Hy:=Matrix(xperp(y*A));
976  Cy:=Hy*A*Transpose(Hy);
    Ly:=LatticeWithGram(Cy);

    ok,T := IsIsometric(Lx,Ly);

981  if not ok then
        return false, _;
    end if;

    T:=DiagonalJoin(ChangeRing(T,QQ),Matrix(QQ,1,[1]));
986  Hx:=ChangeRing(VerticalJoin(Hx,Matrix(x)),QQ);
    Hy:=ChangeRing(VerticalJoin(Hy,Matrix(y)),QQ);
    TT:=Hy^-1*T*Hx;
    assert TT*ChangeRing(A,QQ)*Transpose(TT) eq ChangeRing(A,QQ);
    Sx:=Trans;
991  ok:=exists(w){TT*ChangeRing(g,QQ) : g in Sx |
                    IsIntegralMatrix(TT*ChangeRing(g,QQ))};
    if ok then
        w:=ChangeRing(w,ZZ);
        assert y*w eq x;
996  assert w*A*Transpose(w) eq A;
        return ok, w;
    else
        print "No integral isometry between ", x,y;
        return ok, _;
1001  end if;
end function;

// returns a D-perfect vector of D-minimum 1 and
1006 // a list of its D-minimal vectors.
function FirstPerfect( A : method:=2, Limit:=10^6,
                      record:=rec<PerfectPointsFormat | >)

    record' ConeTestVectors:=Get_y(A: record:=record);
1011 x:=GetConeVector(A : record:=record);
    xr:=ChangeRing(x,QQ);
    M, mu:=DMinimalVectors(x,A : record:=record);
    MM:=Matrix(M);
    Mperp:=Basis(Nullspace(Transpose(MM)));
1016 y1:=record' ConeTestVectors[1];
    y1r:=ChangeRing(y1,QQ);

```

```

Ar:=ChangeRing(A,QQ);
while not (Mperp eq []) do
  size:=#Mperp;
1021  z:=Vector(Mperp[1]);
      i:=1;

      zr:=ChangeRing(z,QQ);

1026  if xAextr(z,A) le 0 and (zr,y1r) ge 0 then
      z:=-z; // z is not in the closure of the first cone.
      end if;

      if method eq 1 then
1031  x,M,mu:=VoronoiNeighbour1(A,x,z : Limit:=Limit,
                              record:=record);
      else
      x,M,mu:=VoronoiNeighbour2(A,x,z : Limit:=Limit,
                              record:=record);
1036  end if;
      xr:=ChangeRing(x,QQ);
      MM:=Matrix(M);
      Mperp:=Basis(Nullspace(Transpose(MM)));
      assert #Mperp lt size;
1041  end while;
      return 1/mu*ChangeRing(x,QQ), M;
end function;

1046 // This is a copy of the corresponding function in the
// normalizer-package

function OrbitReps(Stab, Dirs)
  R:=IndexedSet(Dirs);
1051  Sx:=GeneratorsSequence(Stab);
      Ps:= [ []^#Sx ];
      for i in [1..#R] do
        for j in [1..#Sx] do
          X:= R[i] * Sx[j];
1056  Append(~Ps[j], Index(R, X));
        end for;
      end for;

      Orbs:= Orbits( sub< Sym(#R) | Ps > );
1061  return R[ [ Rep(x) : x in Orbs ] ];
end function;

intrinsic GetPerfectPoints(A:: Mtrx : Limit:=10^6, max:=Infinity(),
1066  method:=2, PerfectPoints:=rec< PerfectPointsFormat | >
  ->[])
{compute all D-perfect points with D-minimum 1 up to action
 of Aut(A)^tr/<- id>}

1071  if not (IsHyperbolic(A) and IsIntegralMatrix(A)) then
      error "Argument must be an integral hyperbolic matrix.";
      end if;

  A:=ChangeRing(A,ZZ);

```

```

1076     pp:= assigned PerfectPoints'Points select PerfectPoints else
           rec< PerfectPointsFormat | Complete:=false>;

    if assigned pp'Points and
1081       (assigned pp'Complete and pp'Complete) then
       return pp'Points;
    end if;

    pp'ConeTestVectors:=Get_y(A);
1086     x, Mx:=FirstPerfect(A : record:=pp, method:=method, Limit:=Limit);
    print "Got first perfect point. It has ", #Mx, " D-minimal vectors";
    pp'Points:=[x];
    pp'directions:=[Directions(x,A : M:=Mx, record:=pp)];
    print "Found ",#pp'directions[1]," directions of first point";
1091     pp'Lat:=[];
    pp'Stabilizers:=[];
    pp'Connectors:=[];
    pp'Neighbours:=[];

1096     // If there are no directions, then there is just one perfect
    // point and we can leave

    if #pp'directions[1] eq 0 then
      S,_L:=PerfectStab(x,A);
1101     Append(~pp'Stabilizers, S);
      Append(~pp'Lat, L);
      pp'AutGrp:=pp'Stabilizers;
      pp'Complete:=true;
      return pp;
1106     end if;

    inspected:=0;

    while inspected lt #pp'Points and #pp'Points le max do
1111     inspected += 1;
      print "Testing number ",inspected;
      x:=pp'Points[inspected];
      Sx,_Lx:=PerfectStab(x,A);
      Append(~pp'Stabilizers, Sx);
1116     Append(~pp'Lat, Lx);
      for r in OrbitReps(Sx, pp'directions[inspected]) do
        if method eq 1 then
          y,My:=VoronoiNeighbour1(A,x,r : Limit:=Limit, record:=pp);
        else
1121         y,My:=VoronoiNeighbour2(A,x,r : Limit:=Limit, record:=pp);
        end if;
        _,Trans:=PerfectStab(y,A);

        for i in [1..#pp'Points] do
1126         ok,T:=PerfectConnector(y, pp'Points[i],A : Trans:=Trans);
          if ok then
            Append(~pp'Connectors, <inspected, i, y, T>);
            continue r;
          end if;
        end for;
1131     end for;

    Append(~pp'Points,y);

```

```

Append(~pp'Neighbours, <inspected, #pp'Points>);
Append(~pp'directions, Directions(y,A: record:=pp, M:=My));
1136 print #pp'Points, ": Found point with"
      ,#pp'directions[#pp'directions],"directions";
end for;
end while;

1141 pp'Complete:= inspected eq #pp'Points;

if pp'Complete then
  pp'AutGrp:= &cat [GeneratorsSequence(Stab) :
1146     Stab in pp'Stabilizers] cat [T[4] : T in pp'Connectors |
      T[4] ne IdentityMatrix(ZZ, Nrows(A))];
end if;

PerfectPoints:=pp;

1151 return PerfectPoints;
end intrinsic;

// Now put everything together and obtain the automorphism
1156 // group of A (modulo -I_n)

intrinsic AutGroup(A:: Mtrx :
  PerfectPoints:=rec< PerfectPointsFormat | >,
  Watson:=true,Limit:=10^6, max:=Infinity(),
1161  method:=2)
  -> GrpMat
{Calculate the automorphism group of an integral
 hyperbolic lattice represented by its Gram matrix.}

1166 if not (IsHyperbolic(A) and IsIntegralMatrix(A)) then
  error "Argument must be an integral hyperbolic matrix.";
end if;

n:=Nrows(A);
1171 GLn:=GL(n,ZZ);
GLnQ:=GL(n,QQ);
ZZn:=Basis(StandardLattice(n));
ZZn:=[ChangeRing(Vector(1),QQ) : 1 in ZZn];

1176 if (assigned PerfectPoints'Complete and PerfectPoints'Complete) and
  assigned PerfectPoints'AutGrp then
  Gens:=[ChangeRing(g, ZZ) : g in PerfectPoints'AutGrp];
  Aut:=sub< GLn | Gens cat [-GLn!1]>;
else
1181 if Watson then
  B,T:=WatsonProcess(A);
  pp:=GetPerfectPoints(B : method:=method);
  print "Starting stabilizer computation.";
  Gens:=[T^-1*ChangeRing(g,QQ)*T : g in pp'AutGrp];
1186 AutB:=sub< GLnQ | Gens >;
  Stab:=MyStabGens(AutB,action,ZZn);
  Aut:=sub< GLn | [ChangeRing(g,ZZ) : g in Stab] cat [-GLn!1]>;
  assert forall{g : g in GeneratorsSequence(Aut) |
      g*A*Transpose(g) eq A};
1191 else

```

```

    pp:=GetPerfectPoints(A);
    Gens:=[ChangeRing(g,ZZ) : g in pp'AutGrp];
    Aut:=sub< GLn | Gens cat [-GLn!1]>;
  end if;
1196 end if;

  return Aut;
end intrinsic;

1201
intrinsic DVoronoiGraph(A::Mtrx :
    Limit:=10^6, max:=Infinity(),
    method:=2, PerfectPoints:=rec< PerfectPointsFormat | > )
    -> [],[],[]
1206 {Compute the residue class graph  $\Gamma_D/\text{Aut}(A)$  (see
    e.g. [Opg, Def 1.10 and Lemma 2.1]).
    The first output are D-perfect points forming the vertex set of
     $\Gamma_D/\text{Aut}(A)$ , the second gives a list of pairs symbolizing the
    (direct) neighbouring relations between the vertices and the third
1211 one gives the connecting family together with the targets.}

    if not (IsHyperbolic(A) and IsIntegralMatrix(A)) then
        error "Argument must be an integral hyperbolic matrix.";
    end if;
1216
    if assigned PerfectPoints'Complete and PerfectPoints'Complete and
        assigned PerfectPoints'Points and
        assigned PerfectPoints'Neighbours and
        assigned PerfectPoints'Connectors then
1221
        Vertices:=PerfectPoints'Points;
        Edges:=[<Vertices[e[1]],Vertices[e[2]]> :
            e in PerfectPoints'Neighbours];
        Connects:=[<<Vertices[c[1]],Vertices[c[2]]>,c[4]> :
1226             c in PerfectPoints'Connectors];
    else
        pp:=GetPerfectPoints(A : PerfectPoints:=PerfectPoints,
            Limit:=Limit, max:=max, method:=method);
        Vertices:=pp'Points;
1231 Edges:=[<Vertices[e[1]],Vertices[e[2]]> :
            e in pp'Neighbours];
        Connects:=[<<Vertices[c[1]],Vertices[c[2]]>,c[4]> :
            c in pp'Connectors];
    end if;
1236
    return Vertices, Edges, Connects;
end intrinsic;

```

B.2 Erläuterung der Intrinsic

Das hier beschriebene Paket kann unter

<http://www.math.rwth-aachen.de/~nebe/Bach.html>

als Textdatei `AutHyp.m` heruntergeladen werden. Um das jeweilige Paket verwenden zu können gibt man in der laufenden Magma-Sitzung den Befehl

```
Attach("AutHyp.m");
```

ein und kann dann für die laufende Sitzung alle dort vorhandenen Intrinsic verwenden. Alternativ kann man den Pfad der Datei in der so genannten `spec`-Datei von Magma eingeben, was dazu führt, dass das Paket zu Beginn jeder Magma-Sitzung eingebunden wird. Man vergleiche dazu auch die entsprechenden Befehlsreferenzen in [BCP97].

B.2.1 Das Automorphismen-Paket

Die umfassendsten Informationen liefert das Intrinsic

```
GetPerfectPoints(A).
```

Es wird mit einer ganzzahligen, hyperbolischen Matrix A aufgerufen und gibt ein Record, also eine Auflistung verschiedener Magma-Datentypen zurück. Es gibt 4 zusätzliche Parameter,

```
Limit:           Default: 10^6
max:             Default: Infinity()
method:          Default: 2
PerfectPoints:   Default: rec< PerfectPointsFormat | >.
```

`Limit` gibt eine obere Schranke an, wieviele Versuche gebraucht werden dürfen um den Nachbarn eines Punktes zu finden. Mit `max` kann der Benutzer die Anzahl der zu berechnenden perfekten Punkte beschränken. Setzt man den Parameter `method` auf 1, so wird zur Bestimmung des VORONOI-Nachbarn die Funktion `VoronoiNeighbour1` verwendet, ansonsten die Funktion `VoronoiNeighbour2`. Erstere benutzt das in Bemerkung 3.1.17 beschriebene Verfahren, letztere die Methode aus dem Beweis zu Proposition 3.1.11, welche sich in den meisten Beispielen als deutlich schneller erwiesen hat (vgl. Abschnitt 6.1). Der Grund hierfür liegt wohl darin, dass in der ersten Variante aus Bemerkung 3.1.17 häufiger D -minimale Vektoren von Punkten berechnet werden müssen, was sehr zeitaufwändig sein kann.

Der Parameter `PerfectPoints` erlaubt, sobald er anders gesetzt wird, auf schon bekannte Ergebnisse, die der Benutzer eingibt, zurückzugreifen und so ggf. Rechenzeit zu sparen.

Das zurückgegebene Record enthält die Felder

```
ConeTestVectors,
Points,
Lat,
directions,
```

Neighbours,
 Stabilizers,
 Connectors,
 AutGrp,
 Complete

In `ConeTestVectors` werden die verwendeten Kegeltestvektoren y_1 und y_2 (vgl. Lemma 5.1.2) gespeichert.

Die Felder `Points`, `Lat` bzw. `Stabilizers` enthalten ein Vertretersystem D -perfekter Punkte x mit D -Minimum 1, der zugehörigen Gitter $L(x)$ (vgl. Lemma 5.3.4) bzw. ihre Stabilisatoren als Untergruppen von $GL_n(\mathbb{Z})$.

Das Feld `directions` enthält eine Liste von Listen von Vektoren, die jeweils den Richtungen der perfekten Punkte entsprechen. Die im Feld `Neighbours` enthaltenen Tupel ganzer Zahlen entsprechen den direkten Nachbarschaftsbeziehungen im D -VORONOI-Graphen. Das Tupel $\langle i, j \rangle$ ist dabei so zu interpretieren, dass im Graphen die Kante vom i -ten zum j -ten perfekten Punkt existiert.

Ähnlich verhält es sich beim Feld `Connectors`. Dieses enthält eine Liste von Tupeln der Gestalt $\langle i, j, v, C \rangle$. Hierbei sind die ersten beiden Einträge ganze Zahlen, der dritte ein Vektor und der vierte eine Matrix. Die Bedeutung ist wie folgt anzusehen: über das verbindende Element C sind die perfekten Punkte Nummer i und j benachbart. Der direkte Nachbar vom i -ten perfekten Punkt ist der Vektor v . Ist C die Einheitsmatrix, so ist dies so zu interpretieren, dass die Kante (i, j) im vollen D -VORONOI-Graphen, also nicht nur im Restklassengraphen existiert.

Das Feld `AutGrp` schließlich fasst die Erzeuger der Stabilisatoren und die nichttrivialen verbindenden Elemente zu einer Liste zusammen.

Der Wahrheitswert im Feld `Complete` gibt an, ob der Algorithmus vollständig durchlaufen wurde, also alle Einträge gesetzt sind. Dieses Feld sollte im Parameter `PerfectPoints` **nicht** manuell auf `true` gesetzt werden, denn dann gibt das Intrinsic lediglich das eingegebene Record zurück.

Die weiteren Intrinsic

`AutGroup(A)`

und

`DVoronoiGraph(A)`

des Automorphismenpakets fassen im Wesentlichen die Ergebnisse des Intrinsic `GetPerfectPoints` zusammen. `AutGroup` gibt die von den Matrizen im Feld `AutGrp` erzeugte Matrixgruppe zurück, wobei hier der zusätzliche Parameter

`Watson : Default true`

angegeben werden kann, der bestimmt, ob statt der eingegebenen Matrix A die WATSON-reduzierte Matrix (vgl. Bemerkung 2.2.8) für die Berechnung der perfekten Punkte zugrunde gelegt wird und anschließend die Automorphismengruppe von A mittels Bahnenalgorithmus bestimmt wird (vgl. Abschnitt 5.6),

oder ob dem Intrinsic `GetPerfectPoints` A selbst als Argument übergeben wird.

`DVoronoiGraph` gibt im Wesentlichen die Felder `Points`, `Neighbours` und `Connectors` aus. Als Parameter nehmen beide Intrinsic

`PerfectPoints: Default rec< PerfectPointsFormat | >`,

in dem ggf. schon vorhandene Informationen angegeben werden können.

Symbolverzeichnis

$\text{Aut } \mathcal{V}_i^{>0}$	Automorphismengruppe des dualen Kegels \mathcal{V}_i , $\text{Aut } \mathcal{V}_i^{>0} := \{g \in \text{GL}(\mathcal{V}_i) \mid \mathcal{V}_i^{>0}g = \mathcal{V}_i^{>0}\}$, siehe 3.2.1
$\mathcal{D}_D(v_1)$	D -VORONOI-Bereich von v_1 , siehe 3.1.4
$\Delta(L)$	Diskriminantengruppe des Gitters L , vgl. 2.1.5
$\delta(v)$	Nachbarschaft des Knotens v eines Graphen, vgl. Anhang A
$\text{End}_{\text{sym}}(\mathcal{V})$	Raum der symmetrischen Endomorphismen des Vektorraums \mathcal{V}
$\mathcal{F}(G)$	Raum der G -invarianten Formen, siehe 4.1.1
$\mathcal{F}^{\geq 0}(G), \mathcal{F}^{>0}(G)$	Menge der positiv semidefiniten bzw. definiten Elemente in $\mathcal{F}(G)$, siehe 4.1.1
\mathcal{G}	Gruppengraph
$\gamma(L)$	HERMITE-Konstante des Gitters L , vgl. 2.4.1
Grp	Kategorie der Gruppen
ιe	Anfangsknoten der Kante e eines Graphen, vgl. Anhang A
$\kappa(L)$	Kusszahl des Gitters L , vgl. 2.3.1
$\left(\frac{p}{q}\right)$	(erweitertes) LEGENDRE-Symbol
$\mu_D(v_1)$	D -Minimum von v_1 , siehe 3.1.4
\mathbb{P}	Menge der positiven, rationalen Primzahlen
$\partial\mathcal{V}_i^{>0}$	Rand des dualen Kegels $\mathcal{V}_i^{>0}$, siehe 3.1.1
\mathbb{Q}_p	Körper der rationalen p -adischen Zahlen
$\mathbb{R}_{\text{hyp}}^{n \times n}$	Menge der reellen, hyperbolischen $n \times n$ -Matrizen, siehe 2.1.2.
$\mathbb{R}_{\geq 0}^{n \times n}$	Menge der positiv semidefiniten reellen $n \times n$ -Matrizen
$\mathbb{R}_{>0}^{n \times n}$	Menge der positiv definiten reellen $n \times n$ -Matrizen
τe	Endknoten der Kante e eines Graphen, vgl. Anhang A

$\mathcal{V}_1^{>0}, \mathcal{V}_2^{>0}$	duale Kegel in den reellen Vektorräumen \mathcal{V}_1 und \mathcal{V}_2 bezüglich der Abbildung $\sigma : \mathcal{V}_1 \times \mathcal{V}_2 \rightarrow \mathbb{R}$ (linear und nicht ausgeartet in beiden Komponenten), siehe 3.1.1
$\mathcal{V}_i^{\geq 0}$	Abschluss des dualen Kegels $\mathcal{V}_i^{>0}$ in \mathcal{V}_i , siehe 3.1.1
φ_x	induzierte Linearform, vgl. 2.4.1
$\text{Watson}(L)$	WATSON-Gitter des Gitters L , vgl. 2.2.8
\mathbb{Z}_p	Ring der ganzen p -adischen Zahlen
$B_r(x)$	offene Kugel um x mit Radius r
C_n	zyklische Gruppe der Ordnung n
D_{2n}	Diedergruppe der Ordnung $2n$, $D_{2n} = \langle a, b \mid a^n, b^2, a^b * a^{-1} \rangle$
$G_x, \text{Stab}_G(x)$	Stabilisator von x , $G_x := \text{Stab}_G(x) := \{g \in G \mid xg = x\}$
H_n	Standardvertreter der Matrizen hyperbolischer Bilinearformen, $H_n := \text{diag}(-1, 1, \dots, 1) \in \mathbb{R}^{n \times n}$, vgl. 5.1.1
$L(x)$	Gitter orthogonal zu x bezüglich einer hyperbolischen Matrix A , vgl. 5.3.4
$L^\#$	zum Gitter L duales Gitter, vgl. 2.1.4
$M(L)$	Menge der kürzesten Vektoren des Gitters L , vgl. 2.3.1
$M_D(v_1)$	Menge der D -minimalen Vektoren von v_1 , siehe 3.1.4
$N(L), \min L$	Minimum des Gitters L , vgl. 2.3.1
$N_{\mathbb{Z}}(G)$	Normalisator einer unimodularen Gruppe $G \leq GL_n(\mathbb{Z})$ in $GL_n(\mathbb{Z})$, siehe 4.1.1
P_x	$P_x = xx^{tr} \in \mathbb{R}^{n \times n}$ für einen Vektor $x \in \mathbb{R}^n$
S_n	symmetrische Gruppe vom Grad n
${}_B\Phi^B$	GRAM-Matrix der Bilinearform Φ bezüglich der Basis B , vgl. 2.1.2

Stichwortverzeichnis

- D*-Minimum, 22
- D*-VORONOI-Bereich, 23
- D*-VORONOI-Graph, 27
- G*-extrem, 40
- G*-perfekt, 40
- p*-Füllung, 12
- BRAVAIS-Gruppe, 37
- BRAVAIS-Mannigfaltigkeit, 37
- HERMITE-Normalform, 11
- HERMITE-Ungleichung, 14
- JORDAN-Zerlegung, 13
- LOBAČEVSKY-Raum, 49
- VORONOI-Bereich, 17
- VORONOI-Graph, 18
- VORONOI-Klassengraph, 18
- WATSON-Prozess, 13

- adjungiert, 28
- Automorphismengruppe, 28
- Automorphismus, 11

- Baum, 70
- benachbart, 18, 69
- Bilinearform, 9

- Diskriminante, 10
- Diskriminantengruppe, 10
- duale Kegel, 21

- eigentlich diskontinuierlich, 28

- Facette, 18, 25
- Fundamentalebereich, 33
- Fundamentalgruppe, 70

- Geschlecht, 12
- Gitter, 10
 - perfektes, 16
 - duales, 10
 - eutaktisches, 16
 - extremes, 16
 - ganzes, 10
- Graph, 69
 - einfacher, 69
 - endlicher, 69
 - lokal endlicher, 69
 - zusammenhängender, 70
- Gruppe, 5
 - (eigentlich) diskontinuierliche, 28
 - adjungierte, 28
 - unimodulare, 5
- Gruppengraph, 70

- Isometrie, 11

- Kachelung, 27
- Kantengruppe, 70
- Knotengruppe, 70
- Kreis, 70
- Kusszahl, 14

- MKGF-Menge, 43

- Nachbar, 18, 26

- Pfad, 69

- quadratische Form, 12
 - äquivalent, 12
 - diagonalisierbare, 12
 - eutaktische, 16
 - extreme, 16
 - perfekte, 16

- Restklassengraph, 70
- Richtung, 18, 25

- Satz
 - MINKOWSKIScher Gitterpunktsatz, 43
 - SYLVESTERScher Trägheitssatz, 9
 - WITTScher Fortsetzungssatz, 15
 - von BASS und SERRE, 30
 - von KOECHER, 34
 - von KORIN und SOLOTARJOW, 17
 - von LOVÁSZ, 44

von VINBERG, 50
Schleife, 69
Spannbaum, maximaler Teilbaum, 70

Vektor

- D*-minimaler, 23
- D*-perfekter, 23

verbindende Familie, 70

zulässig, 24
Zusammenhangskomponente, 70

Literaturverzeichnis

- [All12] ALLOCK, DANIEL: *The Reflective Lorentzian Lattices of Rank 3*. Memoires of the American Mathematical Society, 2012.
- [BCCZ09] BASU, AMITABH, MICHELE CONFORTI, GÉRARD CORNUÉJOLS und GIACOMO ZAMBELLI: *Maximal Lattice-Free Convex Subsets in Linear Spaces*. Trepper School of Business, (63), 2009.
- [BCP97] BOSMA, WIEB, JOHN CANNON und CATHERINE PLAYOUST: *The Magma algebra system. I The user language*. Journal of Symbolic Computation, 24:235–265, 1997.
- [BM91] BERGÉ, ANNE-MARIE und JACQUES MARTINET: *Réseaux extrêmes pour un groupe d'automorphismes*. Journées arithmétiques (Luminy, 1989), Astérisque, 200:41–66, 1991.
- [BNZ73] BROWN, H., JOACHIM NEUBÜSER und HANS JULIUS ZASSENHAUS: *On Integral Groups. III: Normalizers*. Mathematics of Computation, 27(121):167–182, 1973.
- [Cox34] COXETER, HAROLD SCOTT MACDONALD: *Discrete groups generated by reflections*. Annals of Mathematics, 35(3):588–621, 1934.
- [CS88] CONWAY, JOHN HORTON und NEIL JAMES ALEXANDER SLOANE: *Sphere Packings, Lattices and Groups*. Springer-Verlag, 1988.
- [Dic80] DICKS, WARREN: *Groups, Trees and Projective Modules*. Springer-Verlag, 1980.
- [DKT87] DOMICH, PAUL D., R. KANNAN und LESLIE E. TROTTER: *Hermitic normal form computation using determinant arithmetic*. Mathematics of Operations Research, 12(1):50–59, 1987.
- [GAP] *GAP - Reference Manual*.
- [JC95] JAQUET-CHIFFELLE, DAVID-OLIVIER: *Trois Théorèmes de finitude pour les G -formes*. Journal de Théorie des Nombres de Bordeaux, 7(1):165–176, 1995.
- [Kne02] KNESER, MARTIN: *Quadratische Formen*. Springer-Verlag, 2002.
- [Koe57] KOECHER, MAX: *Positivitätsbereiche im \mathbb{R}^n* . American Journal of Mathematics, 79:575–596, 1957.

- [Koe60] KOECHER, MAX: *Beiträge zu einer Reduktionstheorie in Positivitätsbereichen I*. Mathematische Annalen, 141:384–432, 1960.
- [Koe61] KOECHER, MAX: *Beiträge zu einer Reduktionstheorie in Positivitätsbereichen II*. Mathematische Annalen, 144(2):175–182, 1961.
- [Lor12] LORCH, DAVID: *Einklassige Geschlechter positiv definiter Gitter in Dimension 3*. Diplomarbeit, RWTH Aachen University, 2012.
- [Mar03] MARTINET, JACQUES: *Perfect Lattices in Euclidean Spaces*. Springer-Verlag, 2003.
- [Mc11] MCLEOD, JOHN: *Hyperbolic reflection groups associated to the quadratic forms $-3x_0^2 + x_1^2 + \dots + x_n^2$* . Geometriae Dedicata, 152:1–16, 2011.
- [Neu07] NEUKIRCH, JÜRGEN: *Algebraische Zahlentheorie*. Springer-Verlag, 2007.
- [Opg96] OPGENORTH, JÜRGEN: *Normalisatoren und Bravaismannigfaltigkeiten endlicher unimodularer Gruppen*. Aachener Beiträge zur Mathematik, 1996.
- [Opg01] OPGENORTH, JÜRGEN: *Dual Cones and the Voronoi-Algorithm*. Experimental Mathematics, 10(4):599–608, 2001.
- [OPS98] OPGENORTH, JÜRGEN, WILHELM PLESKEN und TILMAN SCHULZ: *Crystallographic Algorithms and Tables*. Acta Crystallographica, A54:517–531, 1998.
- [Ple] PLESKEN, WILHELM: *Gruppentheorie*. Skript zur Vorlesung, SS 2011 an der RWTH Aachen.
- [PS97] PLESKEN, WILHELM und BERND SOUVIGNIER: *Computing Isometries of Lattices*. Journal of Symbolic Computation, 24:327–334, 1997.
- [PZ89] POHST, MICHAEL und HANS ZASSENHAUS: *Algorithmic algebraic number theory*. Cambridge University Press, 1989.
- [Sag] *Sage's Reference Manual*.
- [Ser80] SERRE, JEAN-PIERE: *Trees*. Springer-Verlag, 1980.
- [Vin72] VINBERG, ERNEST BORISSOWITSCH: *On groups of unit elements of certain quadratic forms*. Mathematics of the USSR Sbornik, 16(1):17–35, 1972.
- [Vin89] VINBERG, ERNEST BORISSOWITSCH: *Linear Representations of Groups*. Birkhäuser-Verlag, 1989.

- [Vor08] VORONOI, GEORGI FEODOSJEWITSCH: *Nouvelles applications des paramètres continus à la théorie des formes quadratiques: 1 Sur quelques propriétés des formes quadratiques positives parfaites.* Journal für die Reine und Angewandte Mathematik, 133:97–178, 1908.
- [Wat62] WATSON, GEORGE LEO: *Transformations of a quadratic form which do not increase the class number.* Proceedings of the London Mathematical Society, 3(12):577–587, 1962.

Eigenständigkeitserklärung

Hiermit versichere ich, Michael Helmut Mertens, geboren am 16. Juli 1989 in Viersen, dass die vorliegende Arbeit von mir selbstständig und ausschließlich unter Zuhilfenahme der angegebenen Quellen und Hilfsmittel verfasst wurde.

Aachen, den 21. September 2012

Michael H. Mertens